

**MENGHITUNG BANYAKNYA ORANG MENGGUNAKAN  
ALGORITMA CONVOLUTIONAL NEURAL  
NETWORK (CNN) BERDASARKAN  
CITRA KEPALA**

**SKRIPSI**

**Oleh :  
MUHAMMAD FAHMI ABIDIN  
NIM. 16650063**



**JURUSAN TEKNIK INFORMATIKA  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM  
MALANG  
2021**

**MENGHITUNG BANYAKNYA ORANG MENGGUNAKAN  
ALGORITMA CONVOLUTIONAL NEURAL  
NETWORK (CNN) BERDASARKAN  
CITRA KEPALA**

**SKRIPSI**

**Diajukan kepada:  
Universitas Islam Negeri Maulana Malik Ibrahim  
Untuk memenuhi Salah Satu Persyaratan dalam  
Memperoleh Gelar Sarjana Komputer (S.Kom)**

**Oleh:  
MUHAMMAD FAHMI ABIDIN  
NIM. 16650063**

**JURUSAN TEKNIK INFORMATIKA  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM  
MALANG  
2021**

## LEMBAR PERSETUJUAN

### MENGHITUNG BANYAKNYA ORANG MENGGUNAKAN ALGORITMA CONVOLUTIONAL NEURAL NETWORK (CNN) BERDASARKAN CITRA KEPALA

#### SKRIPSI

Oleh:  
**MUHAMMAD FAHMI ABIDIN**  
NIM. 16650063

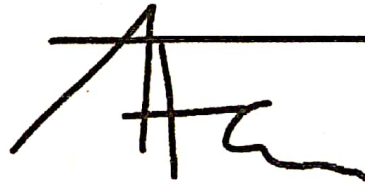
Telah Diperiksa dan Disetujui untuk Diuji  
Tanggal : 4 Juni 2021

Pembimbing I



Irwan Budi Santoso, M.Kom  
NIP. 19770103 201101 1 004

Pembimbing II



Fatchurrochman, M.Kom  
NIP. 19700731 200501 1 002

Mengetahui,  
Ketua Jurusan Teknik Informatika  
Fakultas Sains dan Teknologi  
Universitas Islam Negeri Maulana Malik Ibrahim Malang



  
Cahyo Crysdian  
NIP. 19740424 200901 1 008

## LEMBAR PENGESAHAN

### MENGHITUNG BANYAKNYA ORANG MENGGUNAKAN ALGORITMA CONVOLUTIONAL NEURAL NETWORK (CNN) BERDASARKAN CITRA KEPALA

#### SKRIPSI

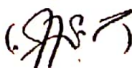
Oleh:  
**MUHAMMAD FAHMI ABIDIN**  
NIM. 16650063

Telah Dipertahankan di Depan Dewan Penguji  
Dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan untuk Memperoleh  
Gelar Sarjana Komputer (S.Kom)  
Pada Tanggal 11 Juni 2021

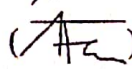
#### Susunan Dewan Penguji

1. Penguji Utama : Dr. M. Amin Hariyadi  
NIP. 19670018 200501 1 001
2. Ketua Penguji : Khadijah Fahmi Hayati Holle, M.Kom  
NIDT. 19900626 20160801 2 077
3. Sekretaris Penguji : Irwan Budi Santoso, M.Kom  
NIP. 19770103 201101 1 004
4. Anggota Penguji : Fatchurrochman, M.Kom  
NIP. 19700731 200501 1 002

()

()

()

()

Mengetahui,  
Ketua Jurusan Teknik Informatika  
Fakultas Sains dan Teknologi  
Universitas Islam Negeri Maulana Malik Ibrahim Malang



Dr. Cahyo Crysdian  
NIP. 19740424 200901 1 008

## PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini :

Nama : Muhammad Fahmi Abidin  
NIM : 16650063  
Fakultas : Sains dan Teknologi  
Jurusan : Teknik Informatika  
Judul Skripsi : Menghitung Banyaknya Orang Menggunakan  
Algoritma *Convolutional Neural Network (CNN)*  
Berdasarkan Citra Kepala

Menyatakan dengan sebenarnya bahwa Skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri, bukan merupakan pengambilalihan data, tulisan atau pikiran orang lain yang saya akui sebagai hasil tulisan atau pikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan pada daftar pustaka.

Apabila di kemudian hari terbukti atau dapat dibuktikan Skripsi ini hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, 2 Juni 2021  
Yang membuat pernyataan,



Muhammad Fahmi Abidin  
NIM. 16650063



## HALAMAN MOTTO

*“Belajarliah dari orang sukses, orang yang putus asa, orang yang takut melangkah dan teruslah mawas diri”*

## HALAMAN PERSEMBAHAN

*Jika ada yang bertanya tentang momen tersulit menjadi mahasiswa? mungkin jawaban saya adalah masa-masa skripsi.*

*Sebagai mahasiswa yang jauh dari kata unggul dan pintar, namun saya bersyukur masih didekatkan dengan orang-orang yang tak kenal lelah dan bosan membantu, mendukung dan memotivasi untuk menghadapi momen tersulit sebagai mahasiswa.*

*“Terimakasih banyak kuhaturkan,  
Semua bukti kasih sayang kalian telah ku abadikan dalam  
lembaran-lembaran ini”*

Untukmu:

*kedua orang tuaku, saudara-saudaraku,  
keluargaku, seluruh dosen dan  
teman seperjuanganku.*

Terima kasih

## KATA PENGANTAR



Assalamu'alaikum. Wr. Wb

Segala puja dan puji syukur penulis haturkan kepada Allah SWT atas Rahmat, Taufik dan Hidayah-Nya penulis dapat menyelesaikan skripsi dengan judul “ **Menghitung Banyaknya Orang Menggunakan Algoritma *Convolutional Neural Network* (CNN) Berdasarkan Citra Kepala**” ini sebagai syarat kelulusan dalam menyelesaikan masa-masa perkuliahan di Universitas Islam Negeri Maulana Malik Ibrahim Malang. Sholawat dan salam semoga tetap terlimpahkan kepada Nabi Muhammad SAW yang telah membimbing dan mewariskan Agama Islam ini sehingga sampai pada sekarang ini.

Keberhasilan penulisan skripsi ini tidak lepas dari dorongan dan bimbingan dari berbagai pihak. Untuk itu dalam kesempatan ini penulis mengucapkan terima kasih yang sebesar-besarnya kepada :

1. **Prof. Dr. Abdul Haris, M.Ag** selaku rektor Universitas Islam Negeri Maulana Malik Ibrahim.
2. **Dr. Sri Hariani, M.Si** selalu dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim.
3. **Bapak Dr. Cahyo Crysdian** selaku Ketua Jurusan Teknik Informatika Universitas Islam Negeri Maulana Malik Ibrahim Malang yang senantiasa memberikan dorongan dan semangat dalam mengerjakan skripsi.



4. **Bapak Irwan Budi Santoso, M.Kom** selaku dosen pembimbing I yang selalu meluangkan waktu dan bersedia memberikan bimbingan serta arahan kepada penulis sehingga dapat menyelesaikan skripsi ini.
5. **Bapak Fatchurrohman, M.Kom** selaku dosen pembimbing II yang juga bersedia meluangkan waktunya dalam membimbing penulis sehingga dapat menyelesaikan skripsi ini.
6. **Bapak** dan **Ibu** berserta **keluarga** yang selalu memberikan kesempatan dan dukungan baik moral, materi maupun spiritual sehingga penulis dapat merasakan pendidikan perkuliyahan sampai akhir.
7. **Seluruh dosen** dan **staff Jurusan Teknik Informatika** yang telah dan sudi menyalurkan ilmu dan pengalamanya.
8. Teman-teman seperjuangan **andromeda**, teman-teman berbagai **komunitas**, teman-teman berbagai **organisasi** dan teman-teman **kontrakan islami** yang telah dan terus memberikan *support* dan mewarnai perjalanan masa-masa perkuliyahan penulis.
9. Semua pihak yang telah membantu dalam penyelesain skripsi ini yang tidak dapat disebutkan satu persatu.

Penulis menyadari sebagai mahasiswa yang jauh dari kata pintar dan cerdas ini, masih terdapat banyak kekurangan dalam penelitian sehingga masih jauh dari kata sempurna dan memuaskan selama penyusunan skripsi ini, oleh karena itu penulis memohon maaf atas segala kekurangan yang terjadi selama proses perkuliyahan dan penyusunan skripsi ini. Semoga tugas akhir ini selalu memberikan kontribusi dan bermanfaat bagi penulis dan pembaca khususnya.

Terakhir salam dari penulis, teruslah semangat dan terus belajarliah dari orang-orang sukses, orang-orang yang telah putus asa, orang yang tidak mau melangkah untuk memulai juga selalu bermawas dirilah.

Malang, 3 Juni 2021

Salam Penulis

## DAFTAR ISI

HALAMAN JUDUL.....	i
LEMBAR PERSETUJUAN.....	ii
LEMBAR PENGESAHAN.....	iii
PERNYATAAN KEASLIAN TULISAN.....	iv
HALAMAN MOTTO.....	v
HALAMAN PERSEMBAHAN.....	vi
KATA PENGANTAR.....	vii
DAFTAR ISI.....	x
DAFTAR GAMBAR.....	xiii
DAFTAR TABEL.....	xv
ABSTRAK.....	xvi
ABSTRACT.....	xvii
المخلص.....	xviii
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang.....	1
1.2. Pernyataan Masalah.....	4
1.3. Tujuan Penelitian.....	4
1.4. Manfaat Penelitian.....	5
1.5. Batasan Masalah.....	5
1.6. Sistematika Penulisan.....	5

BAB II TINJAUAN PUSTAKA.....	7
2.1. Deteksi Objek.....	7
2.2. Menghitung Kerumunan.....	8
2.3. Artificial Neural Network (ANN).....	9
2.4. Deep Learning.....	10
2.5. Convolutional Neural Network.....	11
2.5.1. Convolutional Layer.....	12
2.5.2. Pooling Layer.....	13
2.5.3. Fully Connected Layer.....	14
2.6. Related Work.....	14
BAB III.....	17
METODOLOGI PENELITIAN.....	17
3.1. Desain Sistem.....	17
3.1.1. Input Image.....	19
3.1.2. <i>Extract Region Proposal</i> .....	22
3.1.3. <i>Image Processing</i> Menggunakan CNN.....	29
3.1.4. <i>Training</i> .....	33
3.1.5. <i>Testing</i> .....	33
3.2. Desain Pengujian.....	33
BAB IV UJI COBA DAN PEMBAHASAN.....	38
4.1. Skenario Uji Coba.....	38

4.2.	Hasil Uji Coba.....	38
4.2.1.	Hasil Proses Training.....	39
4.2.2.	Hasil Proses Testing Deteksi Kepala.....	45
4.2.3.	Hasil Proses Testing Menghitung Banyaknya Kepala.....	47
4.3.	Pembahasan.....	55
BAB V PENUTUP.....		62
5.1.	Kesimpulan.....	62
5.2.	Saran.....	62
DAFTAR PUSTAKA.....		64

## DAFTAR GAMBAR

Gambar 2.1 Image Recognition dan Objek Detection.....	7
Gambar 2.2 Artificial Neural Network.....	10
Gambar 2.3 Teknik <i>Pooling</i> .....	14
Gambar 3.1 Perancangan Sistem.....	19
Gambar 3.2 Hasil Treshold Gambar.....	21
Gambar 3.3 Selective Search.....	23
Gambar 3.4 Hasil Segmentasi Gambar.....	24
Gambar 3.5 <i>Bounding Box Region</i> Hasil Segmentasi.....	26
Gambar 3.6 Hasil <i>Region</i> Mengelompokkan <i>Region</i> Yang Berdekatan.....	29
Gambar 3.7 Arsitektur CNN.....	30
Gambar 3.8 <i>Training</i> dan <i>Testing</i> .....	32
Gambar 3.9 5-Fold Cross Validation.....	34
Gambar 4.1 Hasil <i>Training</i> .....	39
Gambar 4.2 Hasil <i>Training</i> Gambar Ukuran 128 dan <i>Epoch</i> 30.....	40
Gambar 4.3 Hasil <i>Training</i> Gambar Ukuran 128 dan <i>Epoch</i> 50.....	40
Gambar 4.4 Hasil <i>Training</i> Gambar Ukuran 128 dan <i>Epoch</i> 75.....	41
Gambar 4.5 Hasil <i>Training</i> Gambar Ukuran 64 dan <i>Epoch</i> 30.....	41
Gambar 4.6 Hasil <i>Training</i> Gambar Ukuran 64 dan <i>Epoch</i> 50.....	42
Gambar 4.7 Hasil <i>Training</i> Gambar Ukuran 64 dan <i>Epoch</i> 75.....	42
Gambar 4.8 Hasil <i>Training</i> Gambar Ukuran 32 dan <i>Epoch</i> 30.....	43
Gambar 4.9 Hasil <i>Training</i> Gambar Ukuran 32 dan <i>Epoch</i> 50.....	43
Gambar 4.10 Hasil <i>Training</i> Gambar Ukuran 32 dan <i>Epoch</i> 75.....	44
Gambar 4.11 Struktur CNN Yang Dibangun.....	48

Gambar 4.12 Hasil Menghitung Orang (8.jpg).....	50
Gambar 4.13 Region Setelah IoU.....	56
Gambar 4.14 Kesalahan Perhitungan.....	57



## DAFTAR TABEL

Tabel 3.1 Hasil Pengaruh Prameter Min Size.....	25
Tabel 3.2 Contoh Perhitungan.....	36
Tabel 4.1 Pengaruh Ukuran Gambar dan Besar Nilai <i>Epoch</i> .....	44
Tabel 4.2 Hasil 5-Fold Cross Validation.....	46
Tabel 4.3 Hasil Perhitungan Orang.....	50

## ABSTRAK

Abidin, Muhammad Fahmi. 2021. **Menghitung Banyaknya Orang Menggunakan Algoritma *Convolutional Neural Network* (CNN) Berdasarkan Citra kepala.** Skripsi. Jurusan Teknik Informatika, Fakultas Sains dan Teknologi. Universitas Islam Negeri Maulana Malik Ibrahim Malang.  
Pembimbing: (I) Irwan Budi Santoso, M.Kom (II) **Fatchurrochman, M.Kom**

---

Kata Kunci : *Convolutional Neural Network*, Menghitung Orang, *CNN*.

Berkumpulnya orang di suatu tempat sehingga membentuk sebuah kerumunan merupakan hal yang lumrah saat ini. Memperkirakan jumlah orang dalam kerumunan merupakan masalah penting untuk berbagai tujuan mulai dari keselamatan umum hingga strategi industri. Misal dalam pengembangan sebuah tempat perbelanjaan, salah satu masalah yang dihadapi pengelola tempat perbelanjaan adalah kurangnya data pengunjung untuk mengetahui jumlah kepadatan pengunjung kios. Data ini sangatlah penting bagi pengelola tempat perbelanjaan untuk mengoptimalkan jam perdagangan juga mengevaluasi daya tarik beberapa area perbelanjaan atau item belanja akan tetapi jika menghitung secara manual akan sulit dan melelahkan bagi manusia. Dalam penelitian ini digunakan metode *convolutional neural network* (CNN) berdasarkan citra kepala orang. CNN merupakan metode yang paling baik dalam melakukan pengenalan dan klasifikasi objek yang terdiri dari beberapa fitur seperti *convolutional layer*, *pooling layer* dan *fully connected layer*. Tujuan dari penelitian ini untuk mengetahui akurasi algoritma CNN dalam menghitung jumlah orang (*people counting*) berbasis citra. Untuk mendeteksi citra kepala pada gambar digunakanya *selective search* untuk mendapatkan *region* gambar yang kemudian digunakan sebagai inputan CNN. Dalam hal ini *region* digunakan untuk mengetahui kemungkinan citra objek pada gambar. Citra kemungkinan objek tersebut dimasukan ke dalam CNN dan dilakukan beberapa skenario uji coba dalam skala ukuran inputan (32x32, 64x64 dan 128x128) dengan masing-masing ukuran inputan *ditraining* dalam besaran nilai epoch (30, 50, 75) sehingga didapatkan hasil (96.28%, 99.24%, 99.49%), (98.39%, 98.73%, 99.58%) dan (95.35%, 98.65%, 99.41%) yang dapat disimpulkan semakin besar skala ukuran inputan maka semakin lama waktu prosesnya juga semakin banyak informasi yang didapat bukan berarti semakin banyak informasi yang didapat membuat sistem semakin akurat. Adapun pengaruh nilai epoch terdapat pada akurasi sitem semakin tinggi nilai epoch maka semakin lama waktu yang dibutuhkan dan akurasi semakin tinggi.

## ABSTRACT

Abidin, Muhammad Fahmi. 2021. **People Counting Using the Convolutional Neural Network (CNN) Algorithm Based on Head Image**. Skripsi. Department of Informatics Engineering, Faculty of Science and Technology. Maulana Malik Ibrahim State Islamic University of Malang.  
Pembimbing: (I) Irwan Budi Santoso, M.Kom (II) **Fatchurrochman, M.Kom**

---

Keyword : *Convolutional Neural Network, People Counting, CNN*

Gathering people in one place to form a crowd is commonplace nowadays. Estimating the number of people in a crowd is an important issue for a variety of purposes from public safety to industrial strategy. For example, in the development of a shopping center, one of the problems faced by shopping mall managers is the lack of visitor data to determine the number of kiosk visitor densities. This data is very important for shopping mall managers to optimize trading hours as well as evaluate the attractiveness of several shopping areas or shopping items, but counting manually will be difficult and tiring for humans. In this research, a convolutional neural network (CNN) method is used based on the image of a person's head. CNN is the best method in recognizing and classifying objects which consist of several features such as a convolutional layer, pooling layer, and fully connected layer. The purpose of this research is to determine the accuracy of the CNN algorithm in counting the number of people (people counting) based on the image. To detect the head image in the image, selective search is used to get the image region which is then used as CNN input. In this case, the region is used to determine the possible image of the object in the image. The image of the possibility of the object is entered into CNN and several test scenarios are carried out in the input size scale (32x32, 64x64, and 128x128) with each input size being trained in epoch values (30, 50, 75) so that the results are (96.28%, 99.24%, 99.49%), (98.39%, 98.73%, 99.58%) and (95.35%, 98.65%, 99.41%) which can be concluded that the larger the input size scale, the longer the processing time also the more information obtained does not mean the more information obtained makes the system more accurate. The effect of the epoch value is on the accuracy of the system, the higher the epoch value, the longer the time required, and the higher the accuracy.

## الملخص

عابدين ومحمد فهمي. 2021. **عد الأشخاص باستخدام خوارزمية الشبكة العصبية التلافيفية (CNN) بناءً على صورة الرأس**. سكريبسي. قسم هندسة المعلوماتية بكلية العلوم والتكنولوجيا. جامعة مولانا مالك إبراهيم الإسلامية الحكومية في مالانج. **بيمينج: (الأول) إيروان بودي سانتوسو ، إم كوم (الثاني) فاتشوروشمان ، إم كوم**

الكلمات المفتاحية: الشبكة العصبية التلافيفية ، عد الأشخاص ، سي إن إن

جمع الناس في مكان واحد لتشكيل حشد هو أمر شائع في الوقت الحاضر. يعد تقدير عدد الأشخاص في حشد ما مسألة مهمة لمجموعة متنوعة من الأغراض من السلامة العامة إلى الاستراتيجية الصناعية. على سبيل المثال ، في تطوير مركز التسوق ، تتمثل إحدى المشكلات التي يواجهها مديرو مراكز التسوق في نقص بيانات الزائرين لتحديد عدد كثافة زوار الأكشاك. تعد هذه البيانات مهمة جدًا لمديري مراكز التسوق لتحسين ساعات التداول بالإضافة إلى تقييم جاذبية العديد من مناطق التسوق أو عناصر التسوق ، ولكن العد يدويًا سيكون صعبًا ومرهقًا للبشر. في هذا البحث ، يتم استخدام طريقة الشبكة العصبية التلافيفية (CNN) بناءً على صورة رأس الشخص CNN. هي أفضل طريقة للتعرف على الكائنات وتصنيفها والتي تتكون من العديد من الميزات مثل الطبقة التلافيفية وطبقة التجميع والطبقة المتصلة بالكامل. الغرض من هذا البحث هو تحديد دقة خوارزمية CNN في حساب عدد الأشخاص (عد الأشخاص) بناءً على الصورة. لاكتشاف صورة الرأس في الصورة ، يتم استخدام البحث الانتقائي للحصول على منطقة الصورة التي يتم استخدامها بعد ذلك كمدخلات CNN. في هذه الحالة ، يتم استخدام المنطقة لتحديد الصورة المحتملة للكائن في الصورة. يتم إدخال صورة إمكانية الكائن في CNN ويتم تنفيذ عدة سيناريوهات اختبار في مقياس حجم الإدخال  $32 \times 32$  و  $64 \times 64$  و  $128 \times 128$  مع تدريب كل حجم إدخال في قيم العصر (30 ، 50 ، 75) بحيث تكون النتائج (99.58% ، 98.73% ، 98.39% ، 99.49% ، 99.24% ، 96.28% و 98.65% ، 95.35%) (99.41% ، والتي يمكن استنتاج أنه كلما كان حجم المدخلات أكبر ، أطول وقت المعالجة أيضًا ، لا تعني المعلومات التي تم الحصول عليها أن المزيد من المعلومات التي تم الحصول عليها تجعل النظام أكثر دقة. يكون تأثير قيمة الحقبة على دقة النظام ، فكلما زادت قيمة الحقبة ، زاد الوقت المطلوب ، وزادت الدقة

# **BAB I**

## **PENDAHULUAN**

### **1.1. Latar Belakang**

Berkumpulnya orang di suatu tempat sehingga membentuk sebuah kerumunan merupakan hal yang lumrah saat ini. Memperkirakan jumlah orang dalam kerumunan merupakan masalah penting untuk berbagai tujuan mulai dari keselamatan umum hingga strategi industri. Misal dalam pengembangan sebuah tempat perbelanjaan, salah satu masalah yang dihadapi pengelola tempat perbelanjaan adalah kurangnya data pengunjung untuk mengetahui jumlah kepadatan pengunjung kios. Data ini sangatlah penting bagi pengelola tempat perbelanjaan untuk mengoptimalkan jam perdagangan juga mengevaluasi daya tarik beberapa area perbelanjaan atau item belanja (Chen & Lin, 2009). Dalam hal ini ketrampilan menghitung jumlah orang dibutuhkan agar dapat memperoleh data dan informasi tentang pengunjung yang akan digunakan sebagai salah satu acuan pengelola tempat perbelanjaan dalam mengambil keputusan yang baik. Akan tetapi jika menghitung secara manual akan sulit dan melelahkan bagi manusia.

Skenario lain misal dalam sebuah komunitas pemrograman web baru saja menyelenggarakan konferensi dengan berbagai sesi yang membahas pemrograman web. Kemudian beberapa panitia di minta untuk menganalisis dan memperkirakan jumlah orang yang menghadiri tiap sesi. Dari analisis tersebut akan digunakan untuk mengevaluasi sesi mana yang menjadi perhatian banyak orang dan sesi mana yang kurang di minati sehingga kedepanya komunitas dapat membuat konferensi yang lebih menarik lagi. Akan tetapi untuk menganalisis jumlah orang dari tiap sesi secara manual akan memakan waktu lama.

Pada dasarnya manusia dapat memprediksi dan menghitung jumlah orang dalam kerumunan secara akurat. Akan tetapi seiring dengan meningkatnya dan besarnya kepadatan kerumunan keakuratan manusia memburuk sehingga tidak dapat diandalkan. Untuk mengatasi masalah tersebut para peneliti sebelumnya telah mengembangkan metode komputer vision dengan berbagai kategori pendekatan seperti pendekatan berbasis deteksi, pendekatan berbasis regresi dan pendekatan berbasis estimasi kepadatan (Wang et al, 2017). Akan tetapi metode-metode tersebut mempunyai banyak tantangan seperti kekacauan yang tinggi, variasi pencahayaan, variasi kepadatan objek, tingginya okulasi dan variasi skala yang disebabkan karena berbagai perspektif. kekacauan tinggi dapat merusak resolusi dari perkiraan peta, dan penerangan cahaya dapat mengurangi akurasinya. Selanjutnya, berbagai kepadatan objek mengurangi akurasi prediksi karena distribusi kepadatan tidak seragam. Demikian pula, oklusi yang parah meningkatkan kesalahan prediksi, dan variasi skala mengurangi prediksi penghitungan dan resolusi kepadatan peta (Ilyas et al, 2020).

Metode berbasis *Convolutional Neural Network* (CNN) baru-baru ini telah mencapai keberhasilan yang signifikan dalam mengatasi beberapa masalah di atas, terutama dalam adegan keramaian kompleks dengan kepadatan tinggi (Sindagi & Patel, 2017). *Convolutional Neural Network* merupakan jenis *neural network* yang khusus untuk memproses data yang memiliki topologi seperti model *grid* contoh data deretan waktu yang bisa dianggap sebagai *grid* 1D yang mengambil sampel interval waktu dengan teratur begitu juga data *image* yang dapat dianggap sebagai *grid* piksel 2D. Nama *convolutional neural network* sendiri menunjukkan bahwa dalam sistem kerja jaringan tersebut memanfaatkan operasi matematika

yang disebut konvolusi. Konvolusi merupakan jenis operasi linier khusus sebagai pengganti perkalian matriks umum setidaknya dalam satu lapisan (Goodfellow et al, 2016).

Berawal dari Krizhevsky et al, (2012) yang melakukan klasifikasi pada 1,2 juta *image* dengan 1000 kelas berbeda yang disediakan ImageNet dengan metode konvolusinya berhasil mencapai *error rate* 15,3%. Ketika CNN berhasil membuktikan kemampuannya, para peneliti mulai mengalihkan perhatiannya untuk mengetahui lebih dari fitur-fitur di dalamnya dan mulai mengembangkannya dalam berbagai permasalahan. Seperti halnya Zhang et al, (2015) mengusulkan metode penghitungan kerumunan berdasarkan CNN pada penelitiannya dan mencapai akurasi yang baik dengan hasil nilai uji MAE (*Mean Absolute Error*) dan MSE (*Mean Square Error*) sebesar 1,60 dan 3,31.

Berdasarkan pemaparan di atas maka, penelitian ini akan mengaplikasikan metode CNN dalam menghitung jumlah orang dengan berdasarkan citra kepala manusia yang terdapat dalam *image*. Diharapkan metode CNN ini dapat memberikan hasil yang sesuai sehingga sistem perhitungan jumlah orang yang akan dibangun pada penelitian ini dapat dimanfaatkan dan dikembangkan kembali dalam hal keamanan, pengembangan bisnis dan berbagai masalah lain yang membutuhkan data jumlah orang dalam kerumunan. Sehingga mempunyai banyak manfaat untuk kemaslahatan orang. Ini sesuai dengan hadits Rasulullah yang diriwayatkan oleh Ahmad, Ath-Thabrani, Ad-Daruqutni dan dihasankan oleh Al-Albani di dalam Shahihul Jami' no 3289 yang berbunyi:

حَيْرُ النَّاسِ أَنْفَعُهُمْ لِلنَّاسِ

Artinya: Sebaik-baik manusia adalah yang paling bermanfaat bagi manusia.



Manusia merupakan makhluk sosial yang mempunyai hubungan timbal balik terhadap manusia satu sama lain sehingga, dalam hal ini Rasulullah menganjurkan umatnya untuk selalu berbuat baik dan bermanfaat bagi manusia khususnya dalam hal kebaikan. Dari hadits diatas dapat disimpulkan bahwa untuk mengetahui tingkat derajat kemuliaan seseorang (khoirunnas) adalah dengan melihat sejauh mana dia memberikan manfaat kepada orang lain yang kehadirannya sangatlah dirindukan karena kepedulian dan kemanfaatan bagi orang lain sehingga, kehadirannya dapat memberikan pengaruh dan energi positif terhadap lingkungan sekitarnya inilah yang disebut dengan (anfauhum linnas).

### 1.2. Pernyataan Masalah

Bermula dari penjelasan pada latar belakang maka didapati pertanyaan yang akan diangkat dalam penelitian ini yaitu:

- a. Seberapa besar akurasi algoritma *convolutional neural network* (CNN) dalam mendeteksi objek kepala pada gambar?
- b. Seberapa akurat algoritma CNN dalam melakukan *people counting* berbasis citra kepala?

### 1.3. Tujuan Penelitian

Pada dasarnya tujuan utama dari penelitian ini adalah untuk mendeteksi dan menghitung jumlah orang pada suatu citra. Adapun maksud dari tujuan ini adalah

- a. mengetahui besar akurasi algoritma CNN dalam mendeteksi objek kepala pada gambar.
- b. mengetahui akurasi algoritma CNN dalam menghitung jumlah orang (*people counting*) berbasis citra.

#### 1.4. Manfaat Penelitian

Manfaat yang diharapkan dari penelitian ini adalah:

- a. Penelitian ini berharap bisa dikembangkan untuk menganalisis jumlah pengunjung untuk menentukan keamanan, strategi penjualan, memperkirakan kepadatan pengunjung dan berbagai hal yang berkaitan dengan data jumlah orang lainnya.
- b. Penelitian ini bisa dikembangkan lagi dan dimanfaatkan pada tempat hiburan dan pusat perbelanjaan.

#### 1.5. Batasan Masalah

Batasan yang ada pada penelitian ini adalah sebagai berikut:

- a. Penelitian ini difokuskan menghitung jumlah orang berdasarkan citra kepala orang pada gambar.
- b. Data yang digunakan pada penelitian ini adalah data berupa foto kepala orang dan foto kumpulan beberapa orang dalam sebuah ruangan.

#### 1.6. Sistematika Penulisan

Penelitian ini tersusun dalam laporan dengan terdiri dari beberapa bab pembahasan sebagai berikut :

- a. Bab I Pendahuluan: pada bab pertama ini berisi tentang latar belakang penelitian, identifikasi masalah, batasan masalah, tujuan penelitian, manfaat penelitian.
- b. Bab II Tinjauan Pustaka: bab ini berisi tentang literature review dan pembahasan teori tentang algoritma *Convolutional Neural Network* (CNN), *object detection*, dll.

- c. Bab III Metode Penelitian: bab ini berisi desain sistem, dan desain pengujian sistem yang dibangun.
- d. Bab IV Hasil dan Pembahasan: bab ini berisi hasil pengolahan sekelompok data *image* orang di dalam ruangan untuk dihitung jumlah orang dalam *image* menggunakan algoritma CNN.
- e. Bab V Penutup: bab ini berisi kesimpulan serta saran yang diharapkan dapat bermanfaat untuk pengembangan penelitian kedepannya.
- f. Daftar Pustaka: berisi daftar referensi penelitian terkait pada penelitian ini.

## BAB II

### TINJAUAN PUSTAKA

#### 2.1. Deteksi Objek

Deteksi objek adalah teknik *computer vision* untuk menemukan objek dalam *image* atau video (Mathwork, 2020) sehingga memungkinkan komputer untuk mengidentifikasi objek dalam *image* atau video. Tujuan deteksi objek adalah untuk mereplikasi kecerdasan manusia kedalam komputer yang dapat mengenali dan menemukan objek pada gambar atau video hanya dalam beberapa saat (Fritz, 2020). Dapat dibilang metode deteksi objek hampir mirip dengan metode *image recognition* hanya saja yang membedakan adalah dalam melabeli objek yang telah diidentifikasi.

*Image recognition* hanya memberikan label pada gambar sedangkan deteksi objek memberikan label objek pada setiap objek yang ada pada gambar. Untuk mempermudah memahami contoh kecil terdapat 2 gambar dengan objek orang untuk gambar pertama terdapat seorang anak laki-laki dan gambar kedua terdapat sekumpulan beberapa orang.



Gambar 2.1 Image Recognition dan Objek Detection

Jika *image recognition* akan memberikan label untuk kedua gambar dengan label orang pada setiap gambar berapapun objek orang yang ada pada gambar sedangkan, deteksi objek akan memberikan label orang pada objek

orangnya saja bukan pada keseluruhan gambar jadi jika pada gambar terdapat 6 orang maka label orang akan disematkan pada setiap objek orang sehingga, deteksi objek dapat memberikan informasi lebih pada gambar dari pada *image recognition*.

Cara kerja deteksi objek secara umum dapat dibagi menjadi dua pendekatan yaitu berbasis *machine learning* dan pendekatan berbasis *deep learning* (Mathworks, 2020). Dalam pendekatan berbasis *machine learning* dengan menggunakan teknik penglihatan komputer untuk melihat berbagai fitur *image* seperti histogram warna atau tepian untuk mengidentifikasi kelompok piksel yang mungkin milik suatu objek. Selanjutnya fitur-fitur tadi dimasukan ke dalam model regresi yang memprediksi lokasi objek beserta labelnya sedangkan, pendekatan *deep learning* menggunakan *convolutional neural network* (CNN) untuk melakukan deteksi objek *end-to-end* tanpa pengawasan. Dimana fitur tidak perlu didefinisikan dan diekstraksi secara terpisah (Fritz, 2020). Oleh karena itu dalam penelitian ini mencoba menggunakan metode *convolutional neural network* dalam melakukan deteksi dan menghitung objek pada *image*.

## 2.2. Menghitung Kerumunan

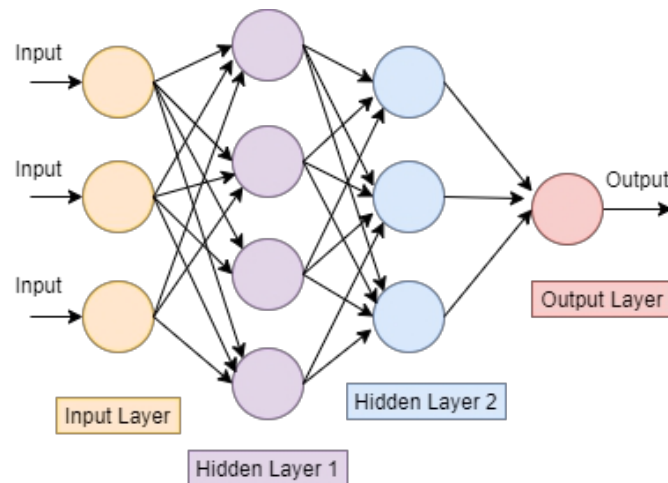
Menghitung kerumunan merupakan tugas yang sulit dikarenakan ada beberapa tantangan yang harus dihadapi seperti kekacauan yang tinggi, variasi pencahayaan, variasi kepadatan objek, tingginya okulasi dan variasi skala yang disebabkan karena berbagai perspektif (Ilyas et al, 2020). Secara umum kerumunan dibagi menjadi 2 yaitu kerumunan padat dan jarang. Dikategorikan sebagai kerumunan padat ketika terdapat banyak orang yang kumpul dan

berdesakan di satu tempat. Begitu juga dikategorikan sebagai kerumunan jarang ketika terdapat kerumunan pada satu tempat tapi tidak berdesak-desakan.

Teknik menghitung kerumunan dibagi menjadi dua kategori dasar yaitu penghitungan kerumunan yang diawasi dan tidak diawasi. Dalam penghitungan kerumunan yang diawasi, data input diketahui dan diberi label dan mesin hanya digunakan untuk menentukan fungsi tujuan (pola tersembunyi). Dalam penghitungan kerumunan tanpa pengawasan data dan label yang digunakan tidak diketahui dan mesin digunakan untuk mengkategorikan dan memberi label data mentah sebelum menentukan fungsi tujuan. Dari kedua teknik tersebut munculah beberapa pendekatan untuk menghitung kerumunan seperti deteksi, regresi, density estimation, convolutional neural network (CNN) dan clustering (Ilyas et al, 2020).

### **2.3. Artificial Neural Network (ANN)**

Artificial Neural Network (ANN) merupakan model komputasi yang terinspirasi dengan karakteristik fungsi otak manusia akan tetapi tidak menggambarkan kompleksitas kinerja otak. Terdapat dua cara kinerja otak yang diadopsi dalam komputasi ANN pertama, blok bangunan yang terdapat pada ANN merupakan perangkat komputasi sederhana yang saling terhubung dengan kuat. Kedua, koneksi antar neuron menentukan fungsi jaringan sehingga, ANN berfungsi sebagai jaringan komputasi terdistribusi paralel dan analog dengan sistem saraf biologis dalam beberapa karakteristik dasar (Park & Lek, 2016). Ada beberapa parameter yang menyusun ANN seperti input layer, hidden layer dan output layer.



Gambar 2.2 Artificial Neural Network

Terlihat gambar 2.2 merupakan contoh dari artificial neural network (jaringan saraf tiruan) yang tersusun oleh input layer yang merupakan lapisan penghubung data inputan ke lapisan selanjutnya kemudian input akan dilanjutkan ke hidden layer 1 pada lapisan ini sistem mempelajari fitur-fitur dasar yang sederhana. Kemudian output dari hidden layer 1 akan digunakan sebagai inputan pada hidden layer 2 yang akan mempelajari fitur yang lebih kompleks dari lapisan sebelumnya demikian pula akan terjadi pada lapisan-lapisan dengan jaringan yang lebih dalam lagi. Kemudian dilanjutkan pada lapisan output. Sehingga dengan cara ini dapat digunakan secara efektif dalam menyelesaikan masalah seperti klasifikasi gambar, pengenalan objek pada gambar dan lainnya.

#### 2.4. Deep Learning

Belakangan ini para peneliti terus mengembangkan metode deep learning hal itu dikarenakan deep learning mempunyai tingkat keakuratan yang tinggi dan cepat dalam melakukan pemrosesan juga seiring dengan berkembangnya perangkat komputasi dengan performa tinggi dan teknologi cloud computing yang keduanya sangat mendukung perkembangan deep learning. Deep learning



merupakan cabang turunan dari machine learning, sesuai dengan namanya deep learning menjadikan komputer untuk belajar lebih dalam dari pengalaman data yang diinputkan. Komputer akan memodelkan hasil pembelajaran dan menyimpannya, fitur inilah yang dimaksud belajar dari pengalaman.

*Deep learning* merupakan sebuah metode pembelajaran dengan berbagai tingkat representasi yang diperoleh dengan menyusun modul sederhana namun *non-linier* yang masing-masing mentransformasikan representasi pada satu tingkat (dimulai dengan input mentah) menjadi representasi pada tingkat yang lebih tinggi (LeCun et al, 2015). Dengan memanfaatkan *deep neural network* yakni kedalaman jaringan saraf dengan menambahkan lebih banyak layer dan lebih banyak unit dalam sebuah layer *deep network* dapat mewakili fungsi-fungsi yang semakin kompleks (Goodfellow et al, 2016). *Deep neural network* telah meningkatkan akurasi mutakhir dalam klasifikasi gambar, deteksi objek, pengenalan suara, terjemahan bahasa, dan bidang lainnya (LeCun et al, 2015).

Salah satu metode *deep learning* yang memanfaatkan *deep neural network* adalah *convolutional neural network* (CNN). Metode ini menggabungkan algoritma konvolusi dengan *neural network* yang merupakan model *deep neural network* yang terbaru dengan pemrosesan data dalam bentuk multi dimensi seperti *image*.

## **2.5. Convolutional Neural Network**

Algoritma *convolutional neural network* (CNN) sangatlah populer belakangan ini. CNN merupakan jenis *neural network* yang khusus untuk memproses data yang memiliki topologi seperti model *grid* contoh data deretan waktu yang bisa dianggap sebagai *grid* 1D yang mengambil sampel interval

waktu dengan teratur begitu juga data *image* yang dapat dianggap sebagai *grid* piksel 2D. Nama *convolutional neural network* menunjukkan bahwa jaringan tersebut operasi matematika yang disebut konvolusi. Konvolusi merupakan jenis operasi linier khusus sebagai pengganti perkalian matriks umum dalam setidaknya satu lapisannya (Goodfellow et al, 2016).

Secara teknis CNN tidak jauh berbeda dengan proses *neural network* lainnya. Terdapat beberapa komponen utama dalam arsitektur CNN yaitu *convolutional layers*, *pooling layers* dan *fully connected layers*.

### 2.5.1. Convolutional Layer

Convolutional layer merupakan blok bangunan utama dari algoritma CNN yang terdiri dari susunan neuron dengan sedemikian rupa yang bertindak sebagai filter. Filter terdiri dari beberapa matriks kecil yang bernilai bobot biasanya disebut dengan kernel kemudian dilakukan proses geser dan dikalikan silang secara berurutan diatas matriks piksel image sehingga menghasilkan feature map. Proses geser itulah yang dinamakan dengan konvolusi. Ada beberapa teknik dalam proses konvolusi sebagai berikut.

- a. *Padding* merupakan teknik konvolusi dengan menambahkan parameter sejumlah pikse disetiap sisi input dengan tujuan untuk memanipulasi dimensi *feature map*.
- b. *Stride* merupakan teknik konvolusi yang menentukan jumlah pergeseran *filter* saat proses konvolusi berlangsung.

Dari kedua teknik tersebut dapat dilakukan secara bersamaan untuk mendapatkan dimensi dan informasi *feature map* yang diinginkan. Secara umum rumus untuk menghitung bentuk matriks output sebagai berikut:

$$\left[ \frac{(n_h - k_h + p_h + s_h)}{s_h} \right] \times \left[ \frac{(n_w - k_w + p_w + s_w)}{s_w} \right] \quad (2.1)$$

Keterangan:

$n_h$  = notasi tinggi dimensi piksel input

$k_h$  = notasi tinggi dimensi piksel kernel

$p_h$  = notasi tambahan *padding* sisi kanan-kiri dimensi *input*

$s_h$  = notasi *stride* vertikal

$n_w$  = notasi lebar dimensi piksel input

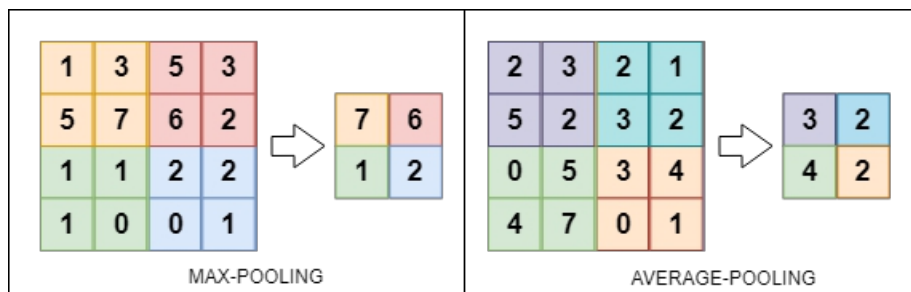
$k_w$  = notasi lebar dimensi piksel kernel

$p_w$  = notasi tambahan *padding* sisi atas-bawah dimensi *input*

$s_w$  = notasi *stride* vertikal

### 2.5.2. Pooling Layer

*Pooling layer* merupakan sebuah *layer* yang terletak setelah *convolutional layer*. Pada dasarnya *pooling layer* terdiri dari sebuah *filter* dengan ukuran dan *stride* tertentu yang akan bergeser pada seluruh area *feature map*. Seperti halnya *kernel* pada proses *filter* di *convolutional layer* hanya saja filter di *layer* ini tidak memiliki nilai bobot dan cukup bekerja dengan membuat lokalitas untuk dilakukan *pooling*. Ada beberapa metode dalam proses *pooling* seperti *max-pooling* dan *average-pooling*.



### Gambar 2.3 Teknik *Pooling*

Pada gambar 2.3 merupakan contoh teknik *pooling* dengan *filter* 2x2 dan *stride* 2 piksel, jika menggunakan teknik *max-pooling* maka setiap pergeseran *filter* 2x2 akan mengambil nilai piksel maksimum. Begitu juga jika menggunakan teknik *average-pooling* maka setiap pergeseran *filter* 2x2 akan mengambil nilai rata-rata pada lokalitas *filter* 2x2. Sehingga tujuan dari *pooling layer* adalah untuk mengurangi dimensi *feature map* dan mempercepat komputasi.

#### 2.5.3. Fully Connected Layer

*Fully connected layer* merupakan lapisan dari *convolutional neural network* yang dibangun dengan jaringan saraf tiruan dengan *feature map* hasil dari proses konvolusi menjadi inputan dari *layer* ini. Dimana seluruh *neuron* pada setiap lapisan jaringan saraf saling terhubung dengan lapisan selanjutnya seperti jaringan saraf tiruan pada umumnya. Sebelum *feature map* digunakan sebagai inputan dari *fully connected layer*, maka akan dilakukan proses *flatten* yaitu proses mengubah dimensi *feature map* yang berdimensi 3D menjadi 1D *list*. Tujuan dari *fully connected layer* ini adalah untuk mengelolah *feature map* sehingga dapat diklasifikasikan.

#### 2.6. Related Work

Beberapa teknik telah digunakan untuk menghasilkan solusi yang tepat untuk mengembangkan pembelajaran mesin dasar dan algoritma visi komputer dalam menghitung kerumunan. Dalam hal ini Chen et al (2012) mengusulkan metode pendekatan berbasis regresi ke berbagai daerah yang dilokalisasi dalam *image* untuk memperkirakan orang di berbagai daerah dengan mempelajari fitur

tingkat rendah seperti detail tepi, piksel latar depan dan output multi-dimensi yang terstruktur akan tetapi mengabaikan informasi spasial yang ada dalam *image*.

Pham et al (2015) mengusulkan untuk menggunakan pendekatan *estimating density* (memperkirakan kepadatan) dengan menggabungkan informasi spasial menggunakan metode *forest regressor* yang digunakan untuk mempelajari pemetaan non-linier. Pemetaan ini berada diantara fitur *patch* dan lokasi relatif semua objek di dalam *patch*. Parameter kepadatan sebelumnya didefinisikan untuk menangani perbedaan antara *patch image* yang padat dan jarang sehingga memberikan dua *forests* yang berbeda.

Akan tetapi baru-baru ini telah ditemukan sebuah metode konvolusi yang memiliki akurasi tinggi dibandingkan dari metode konvensional sebelumnya. Berawal dari Krizhevsky et al (2012) yang melakukan klasifikasi pada 1,2 juta *image* dengan 1000 kelas berbeda yang disediakan ImageNet dengan metode konvolusinya berhasil mencapai *error rate* 15,3%. metode tersebut dikenal dengan AlexNet yaitu sebuah metode yang memanfaatkan algoritma CNN yang menggunakan 5 *convolutional layer* dengan aktivasi ReLu yang diikuti dengan *max-pooling* kemudian diteruskan ke *fully connected layer* yang terdiri dari 3 *layer* dengan 2 *hidden layer* dan softmax.

Wang et al (2015) mengusulkan pendekatan model regresi CNN untuk menghitung jumlah orang. Dalam penelitian tersebut menggunakan algoritma Alexnet untuk memperkirakan jumlah di mana lapisan terakhir pada *fully connected layer* diganti dari 4096 *neuron* dengan satu *neuron* serta menambahkan sampel negatif (*image* yang tidak ada objek orang seperti *image* panorama pepohonan, bangunan, dsb) pada data pelatihanya untuk mengurangi adanya

respon sistem yang menganggap seperti objek pepohonan, bangunan, dsb merupakan objek orang.

Dari kehandalan metode konvolusi yang menghasilkan akurasi tinggi dari pada metode konvensional lainnya maka, peneliti mengusulkan menggunakan metode konvolusi atau biasa dikenal sebagai *convolutional neural network* (CNN) untuk mendapatkan akurasi yang tinggi dalam menghitung banyaknya orang berbasis citra.

## BAB III

### METODOLOGI PENELITIAN

#### 3.1. Desain Sistem

Bab ini menjelaskan tentang tahapan dan perancangan aplikasi penelitian supaya prosesnya dapat terstruktur dengan baik sehingga proses penelitian dapat difahami dan diikuti pihak lain dengan baik. Penelitian ini dirancang untuk menghitung jumlah orang pada citra yang mengimplementasikan algoritma *Convolutional Neural Network* (CNN). Ada beberapa pertanyaan ketika akan membangun sebuah sistem misal seperti apa nantinya sistem yang dibangun? Bagaimana alur prosesnya? Bagaimana cara kerjanya? Oleh karena itu perlu adanya perancangan sistem sehingga dapat menjawab pertanyaan-pertanyaan tersebut dengan begitu sistem akan tergambar dengan jelas. Sistem yang akan dibangun dalam penelitian ini meliputi 5 tahapan yaitu:

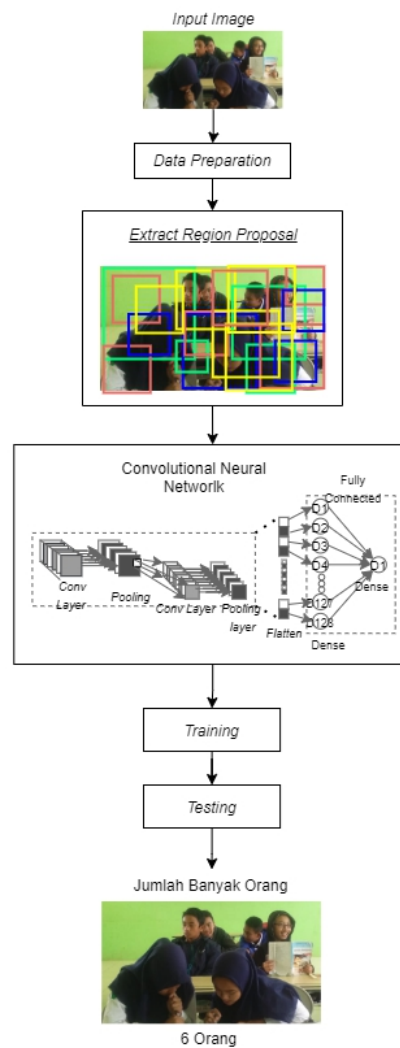
- 1) Tahapan *data preparation* digunakan untuk memanipulasi data.
- 2) *Extract region proposal* merupakan tahapan yang digunakan untuk mengetahui kemungkinan berupa citra objek yang ada pada gambar. Sehingga hasil dari tahapan ini adalah kumpulan berbagai bentuk dan macam citra objek yang ada pada gambar yang kemudian digunakan sebagai inputan pada tahapan *convolutional neural network*.
- 3) Tahapan selanjutnya memodelkan sistem *convolutional neural network* yaitu dengan menentukan berapa banyak *convolutional layer*, *pooling layer*, *filter* dan model *fully connected layer* yang tersusun dari beberapa *layer-layer*, dimana setiap layer terdapat *neuron-neuron* yang saling berhubungan. *Output*



dari tahapan ini adalah berupa jumlah citra kepala manusia yang terdapat pada gambar.

- 4) *Training* merupakan tahapan penting dari keberhasilan sistem. Dengan kata lain jika hasil dari tahapan ini bagus maka kemungkinan besar sistem bekerja dengan baik. *Output* yang dihasilkan dari tahapan proses ini adalah sebuah model *fitting* yang merupakan ciri-ciri atau identitas citra kepala manusia. Model *fitting* ini akan digunakan untuk validasi dan perbandingan bobot dalam proses *testing*.
- 5) *Testing* merupakan tahapan terakhir dalam penelitian ini yang bertujuan untuk mengetahui seberapa baik kinerja sistem dalam mendeteksi dan menghitung jumlah citra kepala manusia pada *image*. Pada tahapan ini akan dilakukan validasi dengan melakukan perbandingan model *fitting* yang didapat dari tahapan *training* sebelumnya.

Secara sederhana perancangan sistem yang akan dibangun pada penelitian ini digambarkan seperti 3.1 berikut:



Gambar 3.1 Perancangan Sistem

### 3.1.1. Input Image

Seperti yang dijelaskan pada latar belakang sebelumnya mengenai studi kasus penelitian ini maka, langkah pertama dalam melakukan penelitian ini yaitu membuat data inputan dengan cara membagi data yang berisikan gambar menjadi 3 macam yaitu:

1. Data *training*, merupakan data yang digunakan untuk proses *learning*.
2. Data *testing*, merupakan data yang digunakan untuk proses *testing*
3. Data *validation*, merupakan data yang digunakan untuk validasi ketika proses *learning* berlangsung.

Tahap awal yaitu menelusuri direktori data yang akan digunakan sistem, dalam direktori dataset terdapat 2 folder data yaitu folder orang yang berisi 500 gambar kepala orang dengan 100 gambar merupakan hasil dari pengambilan gambar berbagai kegiatan menggunakan kamera smartphone dan sisanya merupakan data *self-face-recognition* (didownload dari [www.kaggle.com](http://www.kaggle.com)). Folder satunya merupakan folder bukan orang berisikan 500 gambar ruangan dari dataset *indoor scans* CVPR (didownload dari [www.kaggle.com](http://www.kaggle.com)).

```
nama_file = []
tag = []
dir_file = []
path_d = []
for path, subdirs, files in os.walk(direktori):
    for nama in files:
        dir_file.append(os.path.join(path, nama))
        tag.append(path.split('/')[ -1])
        nama_file.append(nama)
        path_d.append(path)
```

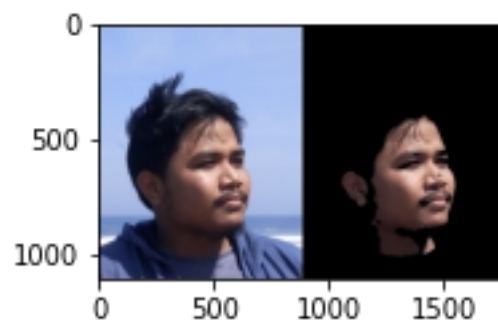
Untuk menelusuri data pada *directory* dan mengambil informasi seperti *path*, *subdirectory* dan *files* dengan memanfaatkan *library* *os.walk*. Informasi yang didapat kemudian disimpan dalam variabel *dir\_file* yang berisi informasi dari *full path* tiap file data, variabel *tag* yang berisi informasi *head path* tiap file, variabel *nama\_file* berisi informasi nama-nama file, *path\_d* berisikan informasi *path* direktori. Kemudian seluruh variabel tadi dimasukan ke *dataframe* agar terlihat rapi dan mudah dibaca dengan memanfaatkan fungsi *dataframe* yang telah disediakan *pandas*.

```
df = pd.DataFrame({"path":dir_file,'nama_file':nama_file,"tag":tag})
```

Tahap selanjutnya yaitu proses thresholding data orang dengan cara mendeteksi kemungkinan yang dianggap warna kulit dan merubah warna selainya dengan warna hitam.

```
def thresholding(gambar):
    gambar_rgb = cv2.cvtColor(gambar, cv2.COLOR_BGR2RGB)
    gambar_ycr = cv2.cvtColor(gambar_rgb, cv2.COLOR_RGB2YCR_CB)
    warna_min = np.array([0,133,77], np.uint8)
    warna_max = np.array([235,173,127], np.uint8)
    kemungkinan_kulit = cv2.inRange(gambar_ycr, warna_min, warna_max)
    hasil = cv2.bitwise_and(gambar_rgb, gambar_rgb,
                           mask = kemungkinan_kulit)
```

Dalam tahapan ini file gambar diubah dahulu menjadi warna YCrCb kemudian menentukan range warna kulit dengan cara menentukan warna minimum dan maksimum kulit dalam ruang warna YCrCb dalam hal ini warna minimum YCrCb dengan *value* [0,133,77] dan untuk warna maksimum YcrCb dengan *value* [235,173,127].



Gambar 3.2 Hasil Threshold Gambar

Berdasarkan gambar 3.2 didapati gambar selain kulit akan diubah menjadi hitam sehingga yang ditampilkan hanya warna yang dianggap sebagai warna kulit saja. Setelah semua file dalam folder orang ditreshold tahapan selanjutnya adalah membagi dataset dari folder orang dan bukan orang.

```
X_train, X_test, y_train, y_test = train_test_split(X, y,
```

```
test_size=0.20, random_state=200)
```

```
X_test, X_val, y_test, y_val = train_test_split(X_test, y_test,
                                              test_size=0.5, random_state=100)
```

Dataset dibagi menjadi 2 dataset dahulu yaitu data *train* dan data *test* dengan perbandingan 20% merupakan data *test* dan 80% merupakan data *train*. Kemudian data *test* dibagi lagi menjadi 2 data yakni data *test* dan data *validation* dengan perbandingan 50% merupakan data *tes* dan 50% sisanya merupakan data *validation*. Sehingga didapat 800 data *train*, 100 data *test* dan 100 data *validation*.

```
if os.path.exists(os.path.join(dataset_path,row['set'],
                                row['tag'])) == False:
    os.makedirs(os.path.join(dataset_path,row['set'],row['tag']))

nama_file_tujuan = file_path.split('/')[-1]
file_tujuan = os.path.join(dataset_path,row['set'],row['tag'],
                            nama_file_tujuan)
```

```
if os.path.exists(file_tujuan) == False:
    shutil.copy2(file_path,file_tujuan)
```

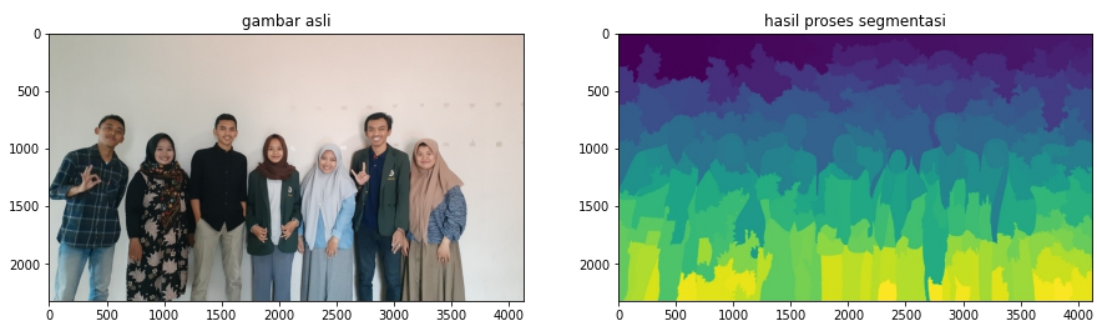
Seluruh data yang sudah dibagi tadi kemudian dimasukkan ke dalam folder *train*, *test* dan valid. Sehingga didapati dari tiap folder (*train*, *test* dan valid) memiliki 2 subfolder yaitu folder orang dan bukan orang.

### 3.1.2. *Extract Region Proposal*

.Untuk mengambil objek yang diinginkan pada *image*, perlu adanya ekstraksi fitur objek sehingga objek dapat dipisahkan dengan *background image*. Untuk melakukan *extraxt region proposal* dalam penelitian ini akan menggunakan *selective search* dengan menggabungkan *exhaustive search* dan segmentasi (metode untuk memisahkan berbagai bentuk objek dalam *image*



Ditahapan ini digunakan *library* yang disediakan oleh *skimage* dengan mencoba mengkonversi piksel gambar menjadi sebuah float dengan menggunakan perintah `skimage.util.img_as_float` yang hasilnya akan digunakan sebagai inputan dari *falzenswalb*. Selanjutnya menghitung segmentasi gambar dengan algoritma *falzenswalb* menggunakan perintah `skimage.segmentation.falzenswalb` yang nantinya akan menghasilkan oversegmentasi gambar multisaluran RGB gambar menggunakan metode *minimum spanning tree* dalam pengelompokan *grid* pada gambar. Parameter *scale* digunakan untuk menentukan tinggi rendahnya luas segmentasi sedangkan parameter *sigma* merupakan diameter *gaussian kernel* yang digunakan untuk *smoothing* gambar sebelum dilakukan proses segmentasi. Parameter *min\_size* digunakan untuk mengontrol ukuran segmentasi gambar. Berikut hasil dari segmentasi *falzenswalb* yang ditunjukkan pada gambar 3.4 di bawah ini:



Gambar 3.4 Hasil Segmentasi Gambar

Untuk mengetahui pengaruh dari perubahan parameter *min\_size* pada pembentukan *region* dari proses segmentasi yaitu dengan mencoba untuk mengganti parameter *min\_size* dengan nilai 50, 500 dan 5000 sehingga hasil yang didapat sebagai berikut:

Tabel 3.1 Hasil Pengaruh Prameter Min Size

<i>min_size</i> = 50	<i>min_size</i> = 500	<i>min_size</i> = 5000
6939 regions	617 regions	60 regions

Dari tabel 3.1 di atas dapat disimpulkan jika semakin kecil nilai *min\_size* maka semakin banyak *region* yang dihasilkan begitu sebaliknya. Dengan begitu semakin banyak *region* yang didapatkan maka, semakin banyak juga info yang bisa didapatkan dari gambar.

Setelah *region* awal sudah terbentuk kemudian dilakukan pengelompokan *region* menggunakan algoritma greedy dengan cara mengelompokkan *region* yang mempunyai kemiripan dan diulang-ulang sehingga seluruh gambar menjadi satu *region* (Uijlings et al, 2013).

```

for y, i in enumerate(img_segment): # panjang vertikal kotak segment
asi
    for x, l in enumerate(i): # panjang horizontal kotak segment
asi

        # inisialisasi region baru
        if l not in region:
            region[l] = {" min_x": np.Inf,
                        " min_y": np.Inf,
                        "max_x": 0,
                        "max_y": 0,
                        "labels": [1]}

        # bounding box
        if region[l][" min_x"] > x:
            region[l][" min_x"] = x
        if region[l][" min_y"] > y:
            region[l][" min_y"] = y

```



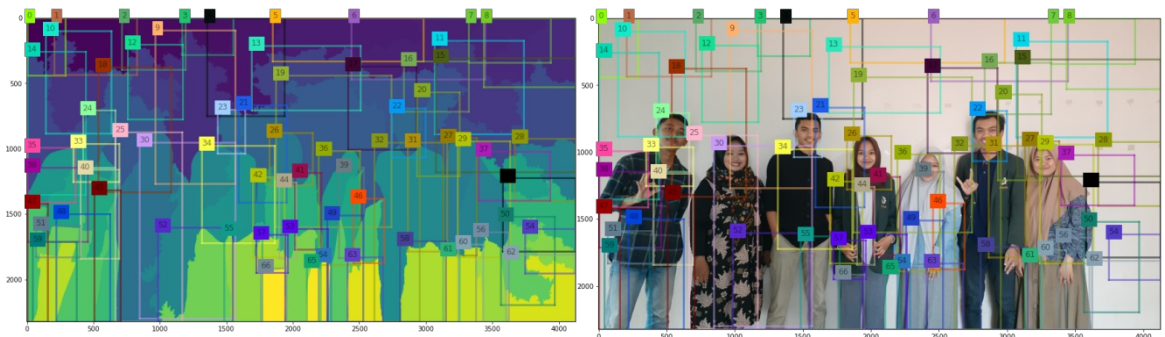
```

if region[1]["max_x"] < x:
    region[1]["max_x"] = x
if region[1]["max_y"] < y:
    region[1]["max_y"] = y

```

Kode tersebut akan menghasilkan *region* berdasarkan segmentasi gambar dengan mengambil informasi panjang dan tinggi *region* gambar dengan *min\_x* dan *max\_x* merupakan panjang *region* sedangkan *min\_y* dan *max\_y* merupakan lebar

dari *region* sehingga membentuk sebuah kotak-kotak *region* yang dinamakan *bounding box* berikut hasil ekstrak *region* yang ditunjukkan pada gambar 3.5.



Gambar 3.5 *Bounding Box Region* Hasil Segmentasi

Pada gambar 3.5 *bounding box* berwarna warni menunjukkan *region* gambar berdasarkan segmentasi gambar.

2) tahapan selanjutnya yaitu mengelompokkan *region* untuk menangkap objek yang lebih detail dengan menggunakan rumus persamaan sebagai berikut:

a. Persamaan warna

$$\text{warna} (ri, rj) = \sum_{k=1}^n \min(c_i^k, c_j^k) \quad (3.1)$$

Keterangan:

$ri, rj$  = pasangan *region*

$n$  = dimensi *region*

$C_i^k, C_j^k$  = warna pasangan region

```
def persamaan_warna(r1, r2):
    return sum([ min(a, b) for a, b in zip(r1["hist_c"],
                                           r2["hist_c"])]])
```

b. Persamaan tekstur

$$tekstur (ri, rj) = \sum_{k=1}^n \min(t_i^k, t_j^k) \quad (3.2)$$

Keterangan:

$ri, rj$  = pasangan region

$n$  = dimensi region

$t_i^k, t_j^k$  = tekstur pasangan region

```
def persamaan_tekstur(r1, r2):
    return sum([ min(a, b) for a, b in zip(r1["hist_t"],
                                           r2["hist_t"])]])
```

c. Persamaan ukuran

$$ukuran (ri, rj) = 1 - \frac{(size(ri) + size(rj))}{size(img)} \quad (3.3)$$

Keterangan:

$size(ri)$  = ukuran region pertama

$size(rj)$  = ukuran region kedua

```
def persamaan_ukuran(r1, r2, imsize):
    return 1.0 - (r1["size"] + r2["size"]) / float(imsize)
```

d. Persamaan bentuk

$$bentuk (ri, rj) = 1 - \frac{(size(BBij) - size(ri) - size(rj))}{size(img)} \quad (3.4)$$

Keterangan:

$\text{size}(r_i)$  = ukuran region pertama

$\text{size}(r_j)$  = ukuran region kedua

$\text{size}(\text{BB}_{ij})$  = ukuran *boundingbox*

```
def persamaan_bentuk(r1, r2, imsize):
    bbsize = ((max(r1["max_x"], r2["max_x"]) - min(r1["
min_x"],
r2[" min_x"]))) * (max(r1["max_y"], r2["max_y"]) -
min(r1[" min_y"], r2[" min_y"])))
    return 1.0 - (bbsize - r1["size"] -
r2["size"]) / imsize
```

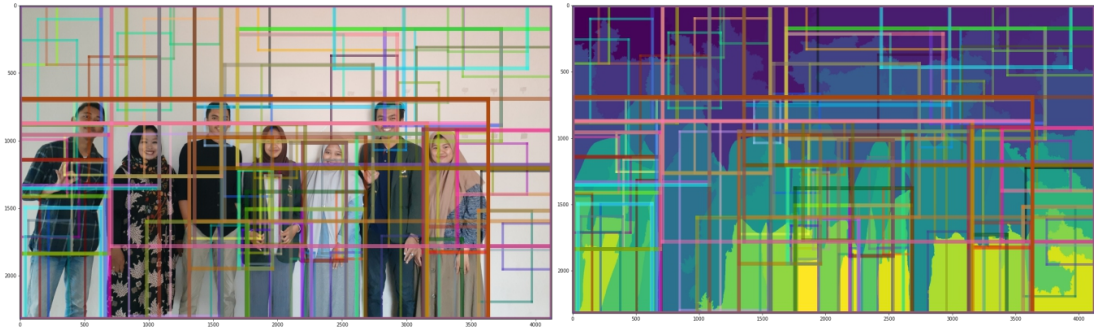
Kemudian menggabungkan semua *region* yang berdekatan sehingga jika ada *region* yang saling berdekatan dan mirip maka akan dikelompokkan dan digabungkan.

```
persamaan_warna(r1, r2) +\
persamaan_tekstur(r1, r2) +\
persamaan_ukuran(r1, r2, imsize) +\
persamaan_bentuk(r1, r2, imsize))
```

Proses pengelompokan *region* yang paling mirip diulang-ulang sampai seluruh *region* bergabung dan menghasilkan satu *region*.

```
for (ai, ar), (bi, br) in region_berdekatan:
    persamaan[(ai, bi)] = menggabungkan_hasil_persamaan(ar,
br, imsize)
```

hasil persamaan tersebut akan menghasilkan *region* baru seperti gambar 3.6 berikut:



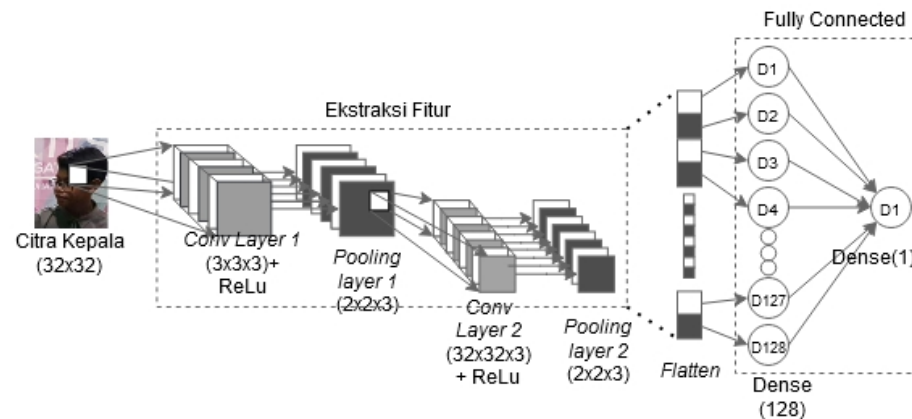
Gambar 3.6 Hasil *Region* Mengelompokkan *Region* Yang Berdekatan

Berdasarkan gambar 3.6 menunjukkan hasil pengelompokan *region* yang berdekatan berdasarkan persamaan warna, tekstur, ukuran dan bentuk. Dengan begitu berbagai bentuk citra objek pada data inputan sudah dikenali akan tetapi belum diketahui apakah citra objek tersebut merupakan citra kepala manusia. Sehingga berbagai bentuk citra objek hasil dari proses ini akan digunakan sebagai inputan tahapan selanjutnya yaitu *convolutional neural network* untuk dilakukan pengenalan dan perhitungan citra kepala manusia.

### 3.1.3. *Image Processing Menggunakan CNN*

Seperti yang telah dijelaskan pada sub bab sebelumnya mengenai CNN maka dalam proses ini sistem akan mengidentifikasi dan menghitung jumlah objek kepala pada *image* menggunakan algoritma CNN. Dengan begitu inputan yang dipakai merupakan objek-objek *image* dengan berbagai bentuk yang didapat dari proses *extract region proposal* sebelumnya.

Mempertimbangkan kinerja CNN yang sangat baik dalam komputer vision maka arsitektur yang akan dibangun pada penelitian ini terdiri dari dua jaringan konvolusi dengan dua *pooling layer* dan dua *fully connected layer*. Secara sederhana gambaran rancangan arsitektur algoritma CNN pada penelitian ini sebagai berikut:



Gambar 3.7 Arsitektur CNN

Untuk memudahkan dalam memahami arsitektur CNN pada gambar 3.7 bisa dijelaskan seperti ini:

1. Inputan citra merupakan citra berbagai bentuk dan ukuran objek hasil proses *extract region proposal* kemudian dilakukan normalisasi citra menjadi 32x32 menggunakan fungsi *rescale* dengan cara membagi piksel objek. Sehingga objek akan mempunyai skala dan ukuran piksel yang sama satu dengan lain dengan ukuran citra objek 32x32x3. maksud dari 32x32x3 adalah citra dengan piksel 32x32 dan angka 3 menunjukkan bahwa citra tersebut merupakan rgb yang kemudian akan dibagi dalam 3 unit saluran yaitu merah, hijau dan biru.
2. Proses konvolusi pertama inputan citra tadi dikalikan silang dengan kernel 3x3x3 dengan *filter* sebanyak 32 dan *stride* sebanyak 1 piksel. Kemudian hasil dari perkalian silang dari masing-masing unit saluran ditambahkan bias kemudian dilakukan perhitungan fungsi aktivasi ReLu.
3. Proses *pooling* pertama yakni membuat area lokalitas dengan cara menggeser *pooling window* pada *feature map* hasil dari proses konvolusi pertama berdasarkan *stride* yang ditentukan sebelumnya. Kemudian dilakukan

pengurangan ukuran *feature map* menggunakan metode *max-pooling* dengan *pooling window* berukuran 2x2 dengan *stride* sebanyak 2 piksel.

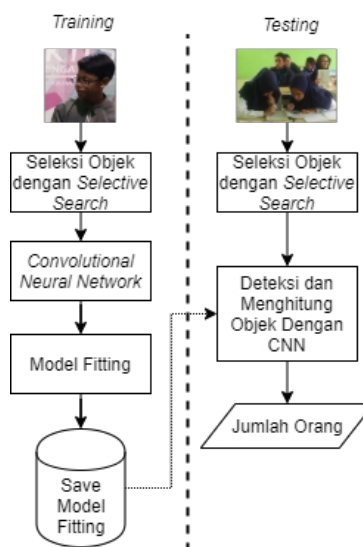
4. Proses konvolusi kedua dengan inputan *feature map* hasil dari *pooling* pertama. Dalam proses konvolusi kedua ini *feature map* akan dikalikan silang dengan *kernel* yang berukuran 3x3x3 sebanyak 32 *filter* dan *stride* sebanyak 1 piksel. Seperti halnya pada proses konvolusi pertama maka dilakukan penambahan bias setiap saluran unit yang kemudian diikuti dengan melakukan fungsi aktivasi menggunakan metode ReLu.
5. Proses *pooling* kedua dengan menggunakan ukuran *pooling window* 2x2 dengan *stride* sebanyak 2 piksel akan dikalikan dengan *map feature* hasil dari konvolusi kedua. Kemudian *feature map* tiap saluran unit hasil proses ini ditambahkan sehingga menjadi satu saluran *feature map*.
6. Proses *flatten* merupakan sebuah proses mengubah *feature map* 3-D menjadi vector 1-D *list*. Dengan begitu *feature map* dapat diteruskan pada proses *fully connected layer*.
7. Proses *fully connected layer* (FC Layer) merupakan jenis jaringan *neural network* pada umumnya. Dalam penelitian ini menggunakan 2 lapisan. Lapisan pertama ditunjukkan dengan *dense* (128) maksudnya adalah terdapat 128 *neuron* pada lapisan tersebut juga untuk metode aktivasi pada lapisan ini menggunakan metode ReLu. Sedangkan lapisan kedua ditunjukkan dengan *dense* (*num\_class*) yang merupakan *output layer* maksudnya *num\_class* adalah banyaknya *dense* tergantung dari banyaknya kelas akan tetapi dalam penelitian ini terdapat 2 kelas yakni objek kepala orang dan objek bukan

kepala orang. Dikarenakan outputnya merupakan 2 maka cukup 2 *neuron* dan menggunakan aktivasi *softmax*.

Berdasarkan gambar 3.6 dapat diimplementasikan kedalam codingan sebagai berikut:

```
cnn = Sequential()
cnn.add(Conv2D(32, (3, 3), padding='same',
               input_shape=input_shape,
               activation = 'relu'))
cnn.add(BatchNormalization())
cnn.add(MaxPooling2D(pool_size = (2, 2)))
cnn.add(Dropout(0.25))
cnn.add(Conv2D(32, (3, 3), activation = 'relu'))
cnn.add(BatchNormalization())
cnn.add(MaxPooling2D(pool_size = (2, 2)))
cnn.add(Dropout(0.5))
cnn.add(Flatten())
cnn.add(Dense(units = 128, activation = 'relu'))
cnn.add(Dense(units = num_class, activation = 'softmax'))
```

Agar Sistem dapat bekerja untuk mengenali dan menghitung orang dalam *image* maka sistem perlu belajar dan dilatih terlebih dahulu dengan dataset *training* dan *testing* yang telah dikumpulkan sebelumnya.



Gambar 3.8 *Training dan Testing*

#### 3.1.4. *Training*

Bagian terpenting dari keberhasilan proses CNN adalah bagusnya hasil dari proses *training* ini. Sistem akan belajar dari dataset *training* untuk mendeteksi dan menghitung citra kepala orang yang terdapat pada *image* seperti yang ditunjukkan pada gambar 3.87 sehingga, menghasilkan model *fitting* berisi model *neuron* yang harus diaktifkan ketika menunjukan citra kepala manusia. Untuk melakukan *update* bobot setiap parameter pada sistem maka digunakan metode optimisasi *Stochastic Gradient Descent* (SGD) dengan mengambil satu sampel dari dataset dan dipilih secara acak untuk setiap iterasi (Geeksforgeeks, 2020). Setelah proses *training* selesai, model *fitting* kemudian disimpan untuk dipanggil kembali pada proses *testing*.

#### 3.1.5. *Testing*

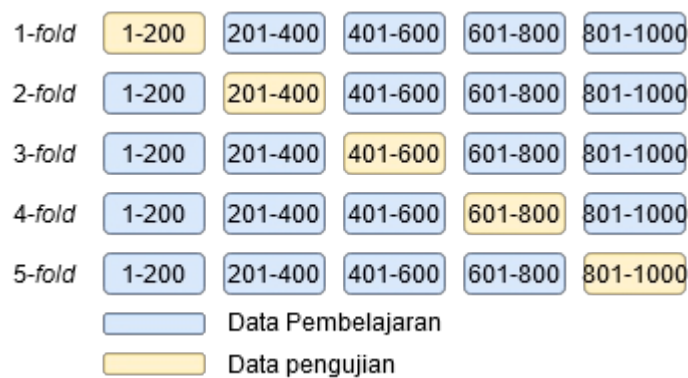
Proses testing dilakukan untuk menguji seberapa baik kinerja model pembelajaran sistem dalam mengidentifikasi dan menghitung jumlah citra kepala manusia pada image. Sistem akan diberi inputan data testing yang belum pernah digunakan sebelumnya. Dalam proses ini model *fitting* yang telah didapat dari proses training akan digunakan sebagai pembanding untuk mengklasifikasi berbagai bentuk citra objek yang ada pada image inputan seperti pada gambar 3.8 dengan dua kategori berupa objek kepala dan bukan kepala. Dengan demikian akan diketahui jumlah citra objek pada gambar yang merupakan citra kepala.

### 3.2. Desain Pengujian

Dalam melakukan prediksi model dan memperkirakan keakuratan sistem yang dibangun pada penelitian ini maka, digunakan metode k-fold cross validation dalam mengelolah data dengan cara membagi dataset menjadi dua



subset yaitu data proses pembelajaran dan data proses validasi. Dataset kemudian dibagi sebanyak k-fold sehingga, 1000 dataset testing dalam penelitian ini akan dibagi menjadi 5 fold dengan 1 fold merupakan subset untuk pengujian dan sisanya digunakan untuk proses pembelajaran. Sehingga terdapat pengulangan proses sebanyak 5 kali dengan subset pembelajaran yang berbeda-beda lihat gambar 3.9 dibawah ini.



Gambar 3.9 5-Fold Cross Validation

Berdasarkan gambar 3.9 dataset yang digunakan untuk pengujian akan bergantian misalnya dalam fold ke-1 subset pengujian ditunjukkan pada data 1-200 sedangkan untuk subset pembelajaran selanjutnya akan bergeser ke sampingnya sehingga pada fold ke-2 subset pembelajaran terletak pada data 201-400 dan begitu seterusnya sampai semua subset pernah digunakan sebagai subset pengujian.

Hasil dari proses pembagian dataset kemudian, dilakukan perhitungan jumlah orang berdasarkan citra kepala seperti halnya yang sudah dijelaskan pada sub bab sebelumnya. Dengan cara mendeteksi citra seluruh objek pada gambar dan dilanjut dengan mengidentifikasi dan mengklasifikasi citra objek ke dalam dua kelas (citra kepala manusia dan bukan citra kepala manusia). Sehingga untuk

menghitung jumlah orang dalam gambar hanya cukup menghitung banyaknya anggota dari kelas citra kepala manusia kemudian, hasil dari perhitungan jumlah orang pada gambar tadi dibandingkan dengan data ground truth jumlah orang pada gambar.

Untuk mengetahui dan mengevaluasi model peramalan menghitung jumlah orang pada gambar dalam penelitian ini menggunakan metode *mean square error* (MSE) dan *mean absolute error* (MAE). Secara matematis MSE dan MAE ditunjukkan sebagai berikut:

$$MAE = \frac{1}{N} \sum_{i=1}^N |C_i - C_i^{GT}| \quad (3.5)$$

$$MSE = \frac{1}{N} \sum_{i=1}^N |C_i - C_i^{GT}|^2 \quad (3.6)$$

$$C_i = \sum_{l=1}^L \sum_{w=1}^W z_{l,w} \quad (3.7)$$

Keterangan:

N = Banyaknya gambar dalam pengujian

$C_i$  = Hasil Penjumlahan matrik *density map*

$C_i^{GT}$  = Hasil penjumlahan matriki ground truth

L = Panjang *density map*

W = Lebar *density map*

$z_{l,w}$  = Piksel pada (l, w) *density map*

Sebagai contoh terdapat 3 citra dengan ukuran 3x3x3 dengan angka 3 terakhir menunjukkan bahwa citra tersebut merupakan rgb sesuai berikut:

Tabel 3.2 Contoh Perhitungan

zl, w	Matriks					
	Ground Truth			Density Map		
	$C_1^{GT}$	$C_2^{GT}$	$C_3^{GT}$	$C_1$	$C_2$	$C_3$
1, 1	2	0	1	1	1	0
1, 2	2	1	1	2	1	2
1, 3	1	2	2	3	2	1
2, 1	4	4	3	3	4	2
2, 2	4	5	5	5	4	3
2, 3	3	2	3	4	2	5
3, 1	3	1	2	2	2	1
3, 2	2	3	0	1	3	1
3, 3	1	0	1	0	1	0
$\Sigma$	22	18	18	21	20	15

Maka:

$$MAE = \frac{1}{N} \sum_{i=1}^N |C_i - C_i^{GT}|$$

$$MAE = \frac{|C_1 - C_1^{GT}| + |C_2 - C_2^{GT}| + |C_3 - C_3^{GT}|}{3}$$

$$MAE = \frac{|21 - 22| + |20 - 18| + |15 - 18|}{3}$$

$$MAE = \frac{(-1) + 2 + (-3)}{3}$$

Kemudian digunakan fungsi absolute sehingga menjadi:

$$MAE = \frac{6}{3}$$

$$MAE = 2$$

Sedangkan perhitungan MSE sebagai berikut:

$$MSE = \frac{1}{N} \sum_{i=1}^N |C_i - C_i^{GT}|^2$$

$$MSE = \frac{|C_1 - C_1^{GT}|^2 + |C_2 - C_2^{GT}|^2 + |C_3 - C_3^{GT}|^2}{3}$$

$$MSE = \frac{|21 - 22|^2 + |20 - 18|^2 + |15 - 18|^2}{3}$$

$$MSE = \frac{(-1)^2 + 2^2 + (-3)^2}{3}$$

$$MSE = \frac{1 + 4 + 9}{3}$$

$$MSE = 4,667$$

Sehingga didapati hasil MAE = 2 dan MSE = 4,667.

Akurasi merupakan kedekatan hasil dari pengukuran dengan data yang sebenarnya. Untuk mengetahui tingkat akurasi model dengan cara membagi jumlah keseluruhan klasifikasi benar dengan jumlah semua dataset awal yang diujikan.

$$Akurasi = \frac{Hasil Benar}{Jumlah Data Testing} \times 100\% \quad (3.8)$$

## BAB IV

### UJI COBA DAN PEMBAHASAN

#### 4.1. Skenario Uji Coba

Penelitian ini melakukan serangkaian uji coba dalam menghitung banyaknya orang dalam sebuah kerumunan menggunakan algoritma CNN menggunakan citra kepala orang yang dapat dilihat sebagai berikut:

- a. Proses *training* merupakan tahapan dalam melatih sistem untuk pengenalan objek pada data yang sudah disiapkan sebelumnya menggunakan model CNN. Kemudian model jaringan yang sudah terlatih disimpan untuk digunakan dalam mengukur seberapa bagus algoritma dalam melakukan identifikasi kepala orang.
- b. Proses *testing* deteksi kepala orang dengan data berupa gambar yang terdapat satu citra objek orang. Ini nantinya untuk mengetahui seberapa akurat dalam mendeteksi kepala orang.
- c. Proses *testing* menghitung banyaknya orang dengan menggunakan data gambar yang berisi beberapa citra objek kepala dalam hal ini akan diketahui hasil akurasi perhitungan sistem dalam menghitung banyaknya objek orang dalam gambar.

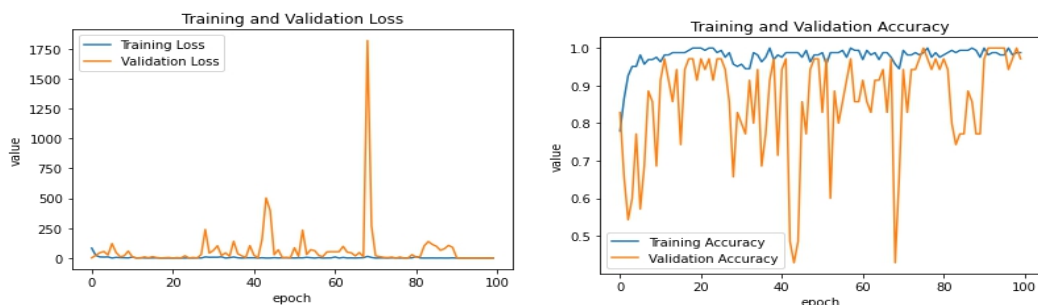
#### 4.2. Hasil Uji Coba

Dalam tahapan uji coba penelitian ini dilakukan pada google colab yang merupakan *tools* seperti Jupyter Notebook yang disediakan oleh google dengan basis *cloud*.

#### 4.2.1. Hasil Proses Training

Tahapan proses *training* menggunakan inputan citra objek hasil dari proses ekstrak *region* untuk pembuatan model pengenalan orang. Dalam proses *training* ini arsitektur CNN yang dibangun seperti yang telah dijabarkan pada sub bab sebelumnya mengenai arsitektur CNN. Dalam proses *training* ini parameter iterasi diberikan nilai sebanyak 100 *epoch* sehingga proses pelatihan akan diulang-ulang sebanyak 100 kali.

Dalam proses ini parameter *batch\_size* digunakan dengan memberikan nilai sebesar 64 yang berarti semua data set akan diambil sebanyak kelipatan 64 data. Misalnya terdapat 1000 data maka untuk *epoch* pertama diambil data mulai dari data ke-0 hingga data ke-63 kemudian untuk *epoch* kedua maka diambil data ke-64 hingga data ke-127 begitu seterusnya sampai pada *epoch* terakhir sisa dari pembagiannya. Sehingga terdapat 16 *batch* dalam satu *epoch*. Maka, total *batch* yang akan ditraining terdapat 800 *batch*. Kemudian nilai parameter *learning rate* ditentukan sebesar 0,01 yang akan berfungsi untuk melakukan *update* bobot pada proses *backward-pass*. Dari 1000 data dengan ukuran (256, 256) akan dilakukan proses *training* sehingga didapatkan hasil seperti gambar 4.1 dibawah ini:

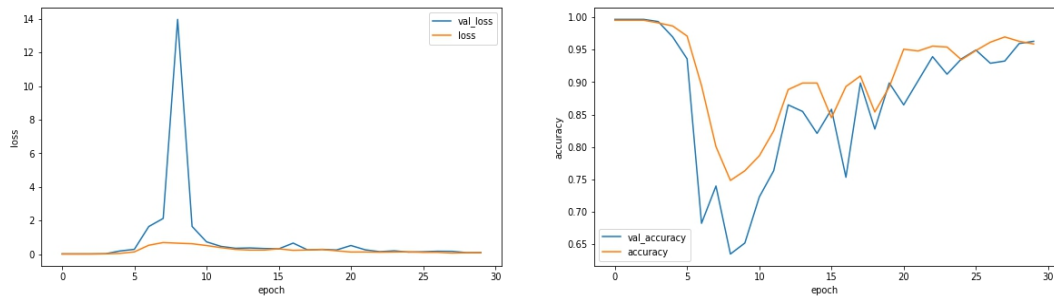


Gambar 4.1 Hasil *Training*

Untuk mengetahui tinggi rendahnya akurasi yang didapat maka dicoba untuk mengganti nilai parameter *learning* dalam membuat model pembelajaran

baru dengan mengubah-ubah ukuran gambar menjadi 3 ukuran (128, 128), (64, 64) dan (32, 32). Juga mengganti nilai epoch dari masing-masing ukuran gambar dengan 30 *epoch*, 50 *epoch* dan 75 *epoch*. Sehingga didapatkan hasil *training* sebagai berikut.

Gambar Inputan (128, 128) Dengan *Epoch* 30



Gambar 4.2 Hasil *Training* Gambar Ukuran 128 dan *Epoch* 30

19/19 [=====] - 6s 306ms/step - loss:

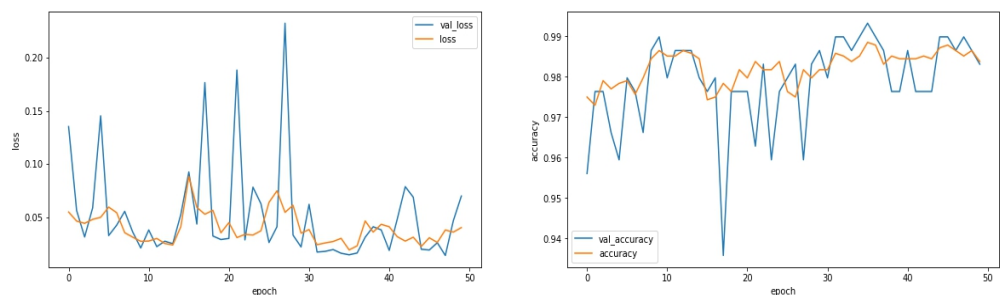
ss0.1188 - accuracy: 0.9535

Tes Akurasi = 95.35%

Tes loss = 0.12

Waktu Yang Dibutuhkan: 16.5621 MIN

a. Gambar (128, 128) dengan *epoch* 50



Gambar 4.3 Hasil *Training* Gambar Ukuran 128 dan *Epoch* 50

19/19 [=====] - 6s 304ms/step - loss:

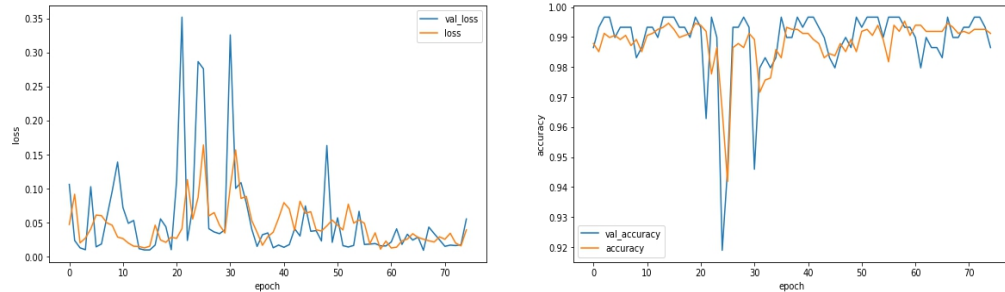
0.0316 - accuracy: 0.9865

Tes Akurasi = 98.65%

Tes loss = 0.03

Waktu Yang Dibutuhkan: 27.8103 MIN

b. Gambar (128, 128) dengan *epoch* 75



Gambar 4.4 Hasil *Training* Gambar Ukuran 128 dan *Epoch* 75

19/19 [=====] - 6s 315ms/step - loss:

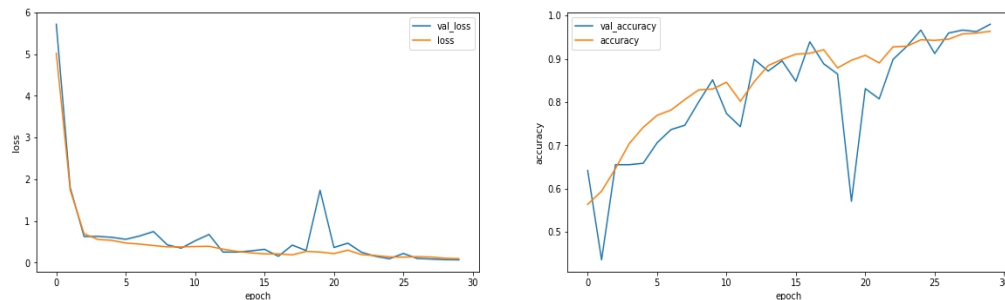
0.0223 - accuracy: 0.9941

Tes Akurasi = 99.41%

Tes loss = 0.02

Waktu Yang Dibutuhkan: 41.7353 MIN

c. Gambar (64, 64) dengan *epoch* 30



Gambar 4.5 Hasil *Training* Gambar Ukuran 64 dan *Epoch* 30

19/19 [=====] - 1s 78ms/step - loss:

0.0580 - accuracy: 0.9839

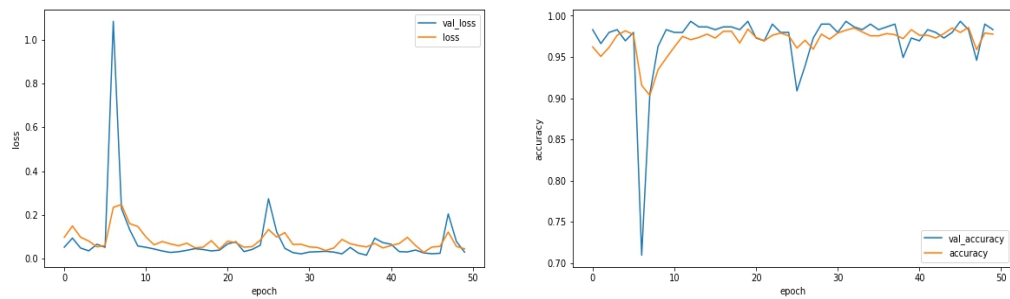
Tes Akurasi = 98.39%

Tes loss = 0.06

Waktu Yang Dibutuhkan: 4.1890 MIN



d. Gambar (64, 64) dengan *epoch* 50



Gambar 4.6 Hasil *Training* Gambar Ukuran 64 dan *Epoch* 50

19/19 [=====] - 1s 78ms/step - loss:

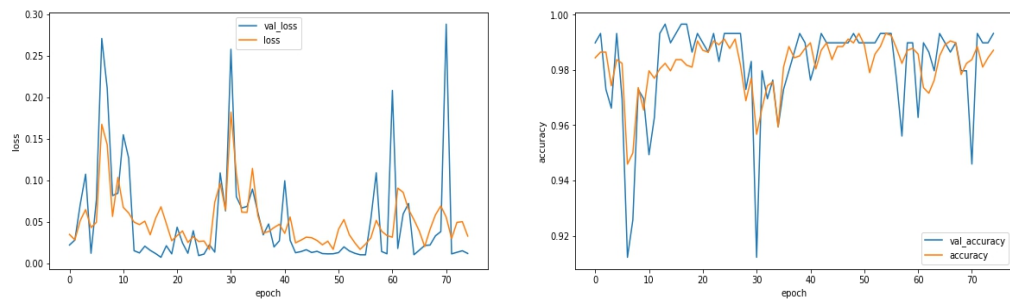
0.0706 - accuracy: 0.9873

Tes Akurasi = 98.73%

Tes loss = 0.07

Waktu Yang Dibutuhkan: 6.9298 MIN

e. Gambar (64, 64) dengan *epoch* 75



Gambar 4.7 Hasil *Training* Gambar Ukuran 64 dan *Epoch* 75

19/19 [=====] - 1s 77ms/step - loss:

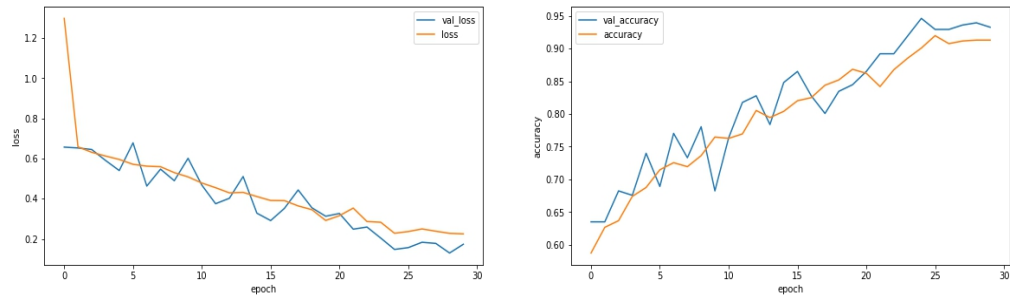
0.0173 - accuracy: 0.9958

Tes Akurasi = 99.58%

Tes loss = 0.02

Waktu Yang Dibutuhkan: 10.3218 MIN

f. Gambar (32, 32) dengan *epoch* 30



Gambar 4.8 Hasil *Training* Gambar Ukuran 32 dan *Epoch* 30

37/37 [=====] - 0s 12ms/step - loss:

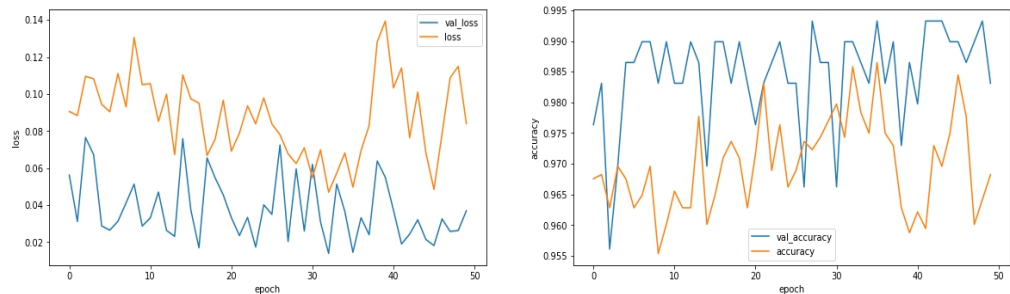
0.1254 - accuracy: 0.9628

Tes Akurasi = 96.28%

Tes loss = 0.13

Waktu Yang Dibutuhkan: 1.0455 MIN

g. Gambar (32, 32) dengan *epoch* 50



Gambar 4.9 Hasil *Training* Gambar Ukuran 32 dan *Epoch* 50

19/19 [=====] - 0s 23ms/step - loss:

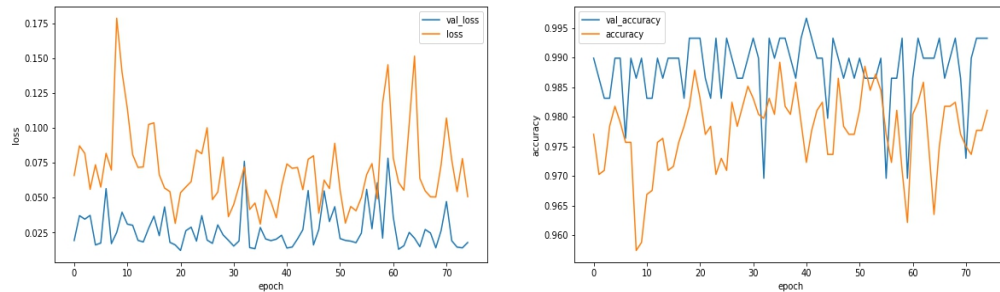
0.0267 - accuracy: 0.9924

Tes Akurasi = 99.24%

Tes loss = 0.03

Waktu Yang Dibutuhkan: 1.7779 MIN

h. Gambar (32, 32) dengan *epoch* 75



Gambar 4.10 Hasil *Training* Gambar Ukuran 32 dan *Epoch* 75

19/19 [=====] - 0s 21ms/step - loss:

0.0114 - accuracy: 0.9949

Tes Akurasi = 99.49%

Tes loss = 0.01

Waktu Yang Dibutuhkan: 2.5821 MIN

Berdasarkan gambar 4.2 sampai gambar 4.10 dapat diketahui adanya pengaruh ketika mengubah ukuran gambar dan nilai *epoch training* untuk memudahkan dalam membacanya maka dirangkum dalam tabel 4.1 dibawah ini:

Tabel 4.1 Pengaruh Ukuran Gambar dan Besar Nilai *Epoch*

Jumlah Epoch	Akurasi	Loss	Waktu Yang Dibutuhkan
Ukuran Gambar (32, 32)			
30	96.28%	0.13	1 Menit
50	99.24%	0.03	1 Menit 42 Detik
75	99.49%	0.01	2 Menit 36 Detik
Ukuran Gambar (64, 64)			
30	98.39%	0.06	4 Menit 12 Detik
50	98.73%	0.07	6 Menit 54 Detik
75	99.58%	0.02	10 Menit 18 Detik

Ukuran Gambar (128, 128)			
30	95.35%	0.12	16 Menit 36 Detik
50	98.65%	0.03	27 Menit 48 Detik
75	99.41%	0.02	41 Menit 42 Detik

Sehingga dapat disimpulkan bahwa ukuran gambar dan besarnya *epoch* sangat mempengaruhi dalam hasil akurasi dan komputasi. Semakin besar gambar maka semakin lama juga komputasinya ini dikarenakan jika ukuran gambar besar maka akan banyak fitur yang diambil begitu sebaliknya. Ini dapat dilihat pada tabel 4.1 ketika ukuran gambarnya berbeda dengan nilai *epoch* yang sama maka semakin banyak waktu yang dibutuhkan.

Sedangkan *epoch* dalam sebuah jaringan CNN sangat berpengaruh dalam hal akurasi dikarenakan banyaknya *batch* yang akan diproses seperti yang dijelaskan pada sub bab sebelumnya mengenai hasil proses training. Hal ini dapat dilihat ketika ukuran gambarnya sama dengan *epoch* yang berbeda maka akan ada perubahan hasil akurasinya

#### 4.2.2. Hasil Proses Testing Deteksi Kepala

Untuk menguji seberapa akurat dalam mendeteksi citra kepala orang maka digunakan metode *5-fold cross validation* dengan cara membagi data menjadi 5 subset data dengan 1 subset digunakan untuk data *training* seperti yang telah dijelaskan pada sub bab sebelumnya.

```
kfold = StratifiedKFold(splits=5, shuffle=True, random_state=seed)
```

Berawal dari membagi data dengan memanfaatkan *method StratifiedKFold* yang disediakan oleh *sklearn* dengan parameter *splits* yang digunakan untuk membagi dataset menjadi 5 subset kemudian data tersebut akan mengacak data sebelum membagi menjadi beberapa kelompok dengan menggunakan parameter *shuffle* akan tetapi bukan berarti mengacak data saat data sudah dikelompokkan dan digunakan dalam proses evaluasi. parameter *random\_state* digunakan untuk mengontrol keacakan yang dilakukan oleh parameter *shuffle*.

Setelah data dibagi langkah selanjutnya adalah menghitung tingkat akurasi model yang dibangun menggunakan *method evaluate* dengan cara memanggil langsung *method* tersebut .

```
evaluasi_model = cnn.evaluate(x=X_train, y=y_train, batch_size=64)
```

Maka didapat hasil pengujian *5-fold cross validation* pada tabel 4.2 sebagai berikut:

Tabel 4.2 Hasil 5-Fold Cross Validation

K-Fold Cross Validation	Hasil Akurasi
Fold-1	91.67%
fold-2	95.65%
Fold-3	96.77%
fold-4	95.65%
fold-5	94.09%
Mean	94.76%

Dapat dilihat pada tabel 4.2 dari hasil pengujian *5-fold cross validation* dengan *fold -1* didapat hasil akurasi sebesar 91,67%, hasil *fold-2* sebesar 95,65%, *fold-3* sebesar 96,77%, *fold-4* sebesar 95,65% dan *fold-5* sebesar 94,09% kemudian dilakukan perhitungan rata-rata sehingga didapatkan hasil akurasi rata-rata sebesar 94,76%. Ini merupakan hasil yang bagus dalam melakukan deteksi objek kepala orang dalam gambar

#### 4.2.3. Hasil Proses Testing Menghitung Banyaknya Kepala

Untuk menghitung jumlah orang yang ada di gambar pertama-tama meresize *region* agar sesuai ukuran inputan model *training* yang telah dibuat dengan perintah berikut.

```
X = []
for i, r in enumerate(regions):
    origx , origy , width, height = r["rect"]
    candidate_region = img[origy:origy + height,
                           origx:origx + width]
    img_resize = skimage.transform.resize(candidate_region,
                                           newsize_img)

    X.append(img_resize)
X = np.array(X)
print(X.shape)
return(X)
```

Dalam hal ini seluruh *region* yang sudah diekstrak akan disimpan info tinggi dan lebar *bounding box* setiap *region* pada variabel *candidate\_region* kemudian dilakukan *resize* dengan memanfaatkan *library skimage* yaitu `skimage.transform.resize()` sehingga didapatkan sebuah *region* dengan yang telah ditentukan yaitu ukuran (256, 256).

Memanggil model *training* yang telah disimpan pada direktori yang telah ditentukan.

```
model = load_model(os.path.join(dir_hasil, "model_epoch50.h5"))
model.summary()
prediksi = model.predict(region)
```

Dengan memanggil file model *training* kemudian disimpan pada variabel `model`. Untuk melihat arsitektur yang dibangun dalam model bisa memanfaatkan `method.summary()` yang disediakan oleh keras maka akan ditampilkan hasilnya seperti gambar 4.11 di bawah ini.

Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d_4 (Conv2D)	(None, 254, 254, 32)	896
batch_normalization_4 (Batch Normalization)	(None, 254, 254, 32)	128
max_pooling2d_4 (MaxPooling2D)	(None, 127, 127, 32)	0
conv2d_5 (Conv2D)	(None, 125, 125, 32)	9248
batch_normalization_5 (Batch Normalization)	(None, 125, 125, 32)	128
max_pooling2d_5 (MaxPooling2D)	(None, 62, 62, 32)	0
dropout_2 (Dropout)	(None, 62, 62, 32)	0
flatten_2 (Flatten)	(None, 123008)	0
dense_4 (Dense)	(None, 128)	15745152
dense_5 (Dense)	(None, 1)	129

Total params: 15,755,681  
 Trainable params: 15,755,553  
 Non-trainable params: 128

Gambar 4.11 Struktur CNN Yang Dibangun

Kemudian menjalankan `model.predict(region)` untuk melakukan prediksi objek orang dari *region* gambar. Langkah selanjutnya adalah menghitung banyaknya *region* yang dideteksi sebagai orang dengan *source code* berikut:

```

prob_orang = np.argsort(fitur[:,0])
prob_orang = prob_orang[::-1]

count = 0

for irow in prob_orang:
    r = regions[irow]
    origx , origy , width, height = r["rect"]
    prob = fitur[irow,0]

    if prob >= hasil :
count += 1

```

Dalam hal ini yang menentukan banyaknya orang yaitu kondisi `if prob >= hasil` yang mana variabel `prob` berisikan nilai fitur orang dan variabel `hasil` merupakan nilai prediksi *region* yang dideteksi sebagai orang. Kemudian untuk melihat hasilnya maka dapat di plot dengan menggunakan *source code* berikut:

```

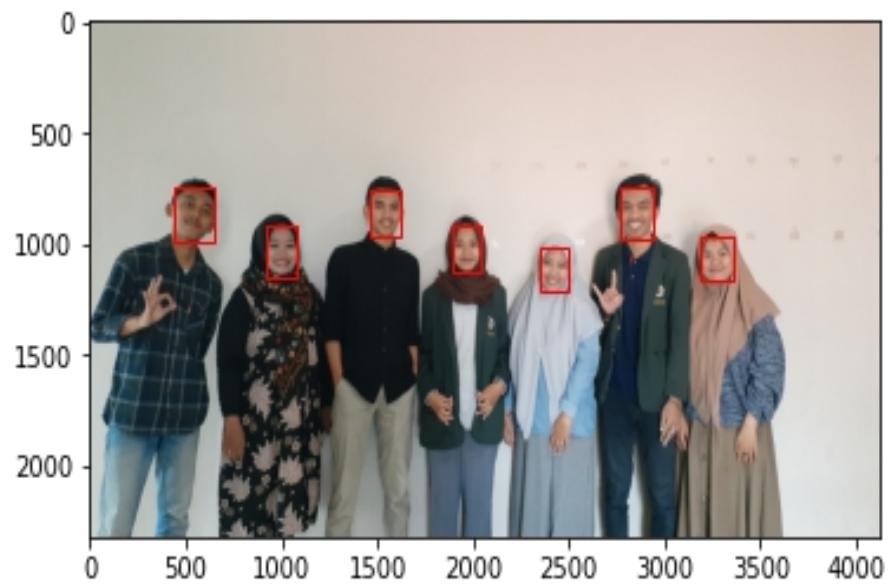
warna = color

plt_kotak(plt,label="orang" + " " + prob.astype(str),
x1 = origx,
y1 = origy,
x2 = origx + width,
y2 = origy + height,
color = warna, alpha=0.5)

```

Sehingga hasil plot dapat ditampilkan seperti gambar 4.12 di bawah ini:





Gambar 4.12 Hasil Menghitung Orang (8.jpg)

Berdasarkan gambar 4.12 kepala orang ditunjukkan dengan *bounding box* berwarna merah. Dalam tes menghitung ini menggunakan 100 data gambar yang diujikan akan keluar hasil prediksi seperti yang ditunjukkan pada tabel 4.3 sebagai berikut:

Tabel 4.3 Hasil Perhitungan Orang

Gambar	Hasil Perhitungan Sistem	Objek Orang Pada Gambar	Keterangan
0.jpg	3	3	Benar
1.jpg	10	8	Salah
2.jpg	7	5	Salah
3.jpg	2	2	Benar
4.jpg	7	7	Benar
5.jpg	4	4	Benar

6.jpg	6	5	Salah
7.jpg	7	7	Benar
8.jpg	7	7	Benar
9.jpg	5	5	Benar
10.jpg	5	5	Benar
11.jpg	6	6	Benar
12.jpg	6	6	Benar
13.jpg	7	6	Salah
14.jpg	6	6	Benar
15.jpg	6	6	Benar
16.jpg	5	5	Benar
17.jpg	4	4	Benar
18.jpg	4	4	Benar
19.jpg	3	3	Benar
20.jpg	4	3	Salah
21.jpg	7	7	Benar
22.jpg	7	7	Benar
23.jpg	3	3	Benar
24.jpg	5	5	Benar
25.jpg	6	6	Benar
26.jpg	11	11	Benar
27.jpg	3	3	Benar
28.jpg	4	4	Benar

29.jpg	4	4	Benar
30.jpg	8	8	Benar
31.jpg	7	7	Benar
32.jpg	13	13	Benar
33.jpg	8	7	Salah
34.jpg	6	6	Benar
35.jpg	11	11	Benar
36.jpg	4	4	Benar
37.jpg	6	6	Benar
38.jpg	13	13	Benar
39.jpg	8	8	Benar
40.jpg	7	7	Benar
41.jpg	5	5	Benar
42.jpg	6	6	Benar
43.jpg	13	12	Salah
44.jpg	7	7	Benar
45.jpg	7	7	Benar
46.jpg	5	5	Benar
47.jpg	5	5	Benar
48.jpg	6	6	Benar
49.jpg	5	5	Benar
50.jpg	12	12	Benar
51.jpg	2	2	Benar

52.jpg	5	5	Benar
53.jpg	7	6	Salah
54.jpg	5	5	Benar
55.jpg	9	7	Salah
56.jpg	10	9	Salah
57.jpg	3	3	Benar
58.jpg	7	5	Salah
59.jpg	11	11	Benar
60.jpg	7	7	Benar
61.jpg	7	6	Salah
62.jpg	10	9	Salah
63.jpg	9	9	Benar
64.jpg	5	5	Benar
65.jpg	5	5	Benar
66.jpg	6	6	Benar
67.jpg	5	5	Benar
68.jpg	5	5	Benar
69.jpg	9	7	Salah
70.jpg	12	12	Benar
71.jpg	2	2	Benar
72.jpg	7	7	Benar
73.jpg	9	9	Benar
74.jpg	5	5	Benar

75.jpg	6	6	Benar
76.jpg	6	6	Benar
77.jpg	14	12	Salah
78.jpg	7	5	Salah
79.jpg	16	14	Salah
80.jpg	9	9	Benar
81.jpg	11	11	Benar
82.jpg	5	5	Benar
83.jpg	5	5	Benar
84.jpg	7	7	Benar
85.jpg	6	6	Benar
86.jpg	9	7	Salah
87.jpg	7	7	Benar
88.jpg	7	7	Benar
89.jpg	13	13	Benar
90.jpg	6	6	Benar
91.jpg	6	6	Benar
92.jpg	6	6	Benar
93.jpg	11	11	Benar
94.jpg	6	6	Benar
95.jpg	11	11	Benar
96.jpg	5	5	Benar
97.jpg	5	5	Benar

98.jpg	5	5	Benar
99.jpg	5	5	Benar

Dari tabel 4.3 didapati menghitung banyaknya orang dengan hasil benar sebesar 82 dan hasil salah sebesar 18 sehingga untuk mengetahui nilai akurasi dari hasil menghitung banyaknya orang dapat dihitung dengan rumus berikut:

$$Akurasi = \frac{Hasil\ Benar}{Jumlah\ Data\ Testing} \times 100\%$$

$$Akurasi = \frac{82}{100} \times 100\%$$

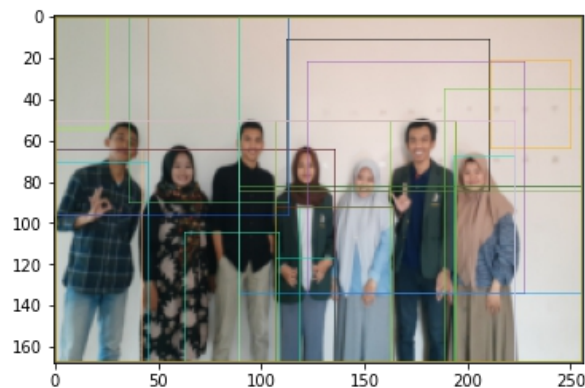
$$Akurasi = 82\%$$

Maka didapat akurasi sebesar 82% dari 100 data tes yang diujikan.

#### 4.3. Pembahasan

Setelah dilakukan uji coba sistem sehingga dapat diketahui bahwa dalam melakukan pemrosesan gambar bagian terpenting dalam sistem ini adalah terletak pada tahap *selective search* dan *training*. Tahap *selective search* disini merupakan tahap yang sangat krusial dikarenakan tahapan ini yang akan menggolongkan data menjadi negatif dan positif dengan cara mengestrak gambar menjadi beberapa region. Dalam sistem ini digunakan 2 kondisi yaitu ketika kondisi *region* 0,7 untuk positif dan kondisi 0,4 untuk negatif. Sehingga akan diketahui *region* yang kemungkinan termasuk objek orang. Ketika masuk dalam tahap *training* data sudah dibagi menjadi 2 yaitu data positif dan negatif yang mana data negatif ini dianggap sebagai *background image*. Akan tetapi kekurangan dari tahapan ini adalah masih banyak *region* yang tumpang tindih setelah dilakukanya IoU sehingga perlu adanya tambahan tahapan untuk meminimalisir tumpang tindihnya

region sehingga dapat meminimalisir waktu yang dibutuhkan juga ketika data *region* banyak yang tumpang tindih saat *training* maka satu *region* objek akan dibaca sebanyak *region* yang menunjukkan objek tersebut seperti yang ditunjukkan gambar 4.13 dibawah ini yang ditunjukkan oleh *bounding box* dengan berbagai warna.

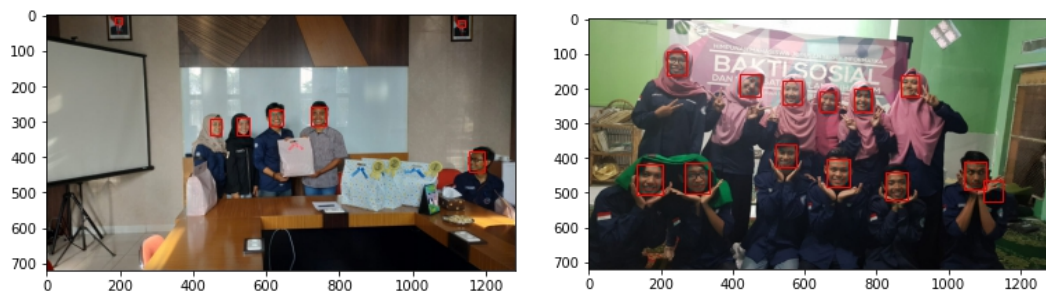


Gambar 4.13 Region Setelah IoU

Dalam tahap *training* menunjukkan angka akurasi yang baik setelah dilakukan pengujian sistem menggunakan *5-fold cross validation* dengan nilai akurasi *fold-1* = 91.67%, *fold-2* = 95.65%, *fold-3* = 96.77%, *fold-4* = 95.65%, *fold-5* = 94.09% dengan *mean* = 94.76% kemudian dikembangkan lagi untuk mengetahui pengaruh dari ukuran gambar dan epoch sehingga dapat diketahui dari tabel 4.1 bahwa semakin besar ukuran gambar yang digunakan sebagai inputan maka waktu yang dibutuhkan semakin lama karena *region* yang dihasilkan semakin banyak juga semakin besar nilai epoch maka didapati nilai akurasi semakin tinggi dikarenakan CNN ini kurang optimal jika data pembelajaranya sedikit.

Terjadinya kesalahan perhitungan sistem dikarenakan masih adanya beberapa *region* yang dianggap sebagai kepala orang semisal kulit tangan

dikarenakan warna kulit tangan hampir sama dengan ciri-ciri kulit wajah selain itu kesalahan perhitungan juga dikarenakan dalam gambar tersebut terdapat foto semisal gambar penyerahan hadiah didalam ruangan dan didalam dinding ruangan tersebut terdapat foto presiden dan wakilnya maka foto tersebut ikut terdeteksi juga padahal yang diinginkan hanyalah pesertanya saja seperti yang ditunjukkan pada gambar 4.14 di bawah ini:



Gambar 4.14 Kesalahan Perhitungan

Pada gambar 4.14 pada sisi kanan merupakan salah satu contoh kesalahan sistem dalam menghitung banyaknya orang pada gambar dan pada sisi kiri merupakan gambar hasil menghitung gambar oleh sistem dengan menganggap foto pada latar belakang merupakan objek orang. Dari tabel 4.3 menunjukkan sekitar 18 kesalahan dari 100 gambar yang diujikan sehingga didapatkan nilai akurasi sebanyak 82% ini menunjukkan hasil yang cukup bagus.

*Convolutional Neural Network* merupakan sebuah metode pemrosesan informasi data (gambar, text dan audio) yang merupakan pengembangan dari jaringan syaraf tiruan dimana jaringan syaraf tiruan sendiri terinspirasi dari sistem kerja sel syaraf manusia dimana letak sel syaraf terpusatnya di otak. Dalam hal ini sel syaraf merupakan suatu sistem yang diciptakan Allah untuk memproses informasi yang diperoleh dari sensor-sensor (indera) pada makhluk hidup dengan syaraf terpusatnya adalah otak sebagai tempat pemrosesan dari



sinyal-sinyal informasi yang diterima juga merupakan pusat kendali instruksi dan respon dari seluruh tubuh makhluk hidup.

Dalam hal ini 14 abad yang lalu Al Qur'an menjelaskan tentang sensor-sensor (indera) pada makhluk dalam surah Al Mu'minun ayat 78:

وَهُوَ الَّذِي أَنْشَأَ لَكُمُ السَّمْعَ وَالْأَبْصَارَ وَالْأَفْئِدَةَ قَلِيلًا مَّا تَشْكُرُونَ

Artinya: “Dan Dialah yang telah menciptakan bagi kamu sekalian, pendengaran, penglihatan dan hati. Amat sedikitlah kamu bersyukur”

Dalam tafsir Jalalain yang dikarang oleh Jalaluddin al-Mahalli & Jalaluddin as-Suyuthi Rohimahullah menjelaskan “(dan Dialah yang menciptakan) yang menjadikan(bagi kamu sekalian pendengaran) lafal “As Sam’u” maknanya Al Asmaa’ dalam bentuk jamak (penglihatan dan kalbu) hati (amat sedikitlah) lafal “maa” mengukuhkan makna yang terkandung dalam lafal “qaliilan” (kalian bersyukur).

Dalam tafsir Al Misbah yang dikarang oleh Muhammad Quraish Shihab menjelaskan “Bagaimana kalian dapat mengingkari Allah padahal Dialah yang menciptakan pendengaran agar kalian dapat mendengar kebenaran, menciptakan penglihatan agar kalian dapat memperhatikan dan mengamati alam raya dengan segala isinya, dan menciptakan pikiran agar kalian dapat mengetahui kemaha agungan- Nya, sehingga kalian beriman? Kalian benar-benar tidak mensyukuri Sang Penciptanya dengan beriman dan [taat](#) kepada-Nya, kecuali sangat sedikit.”

Berdasarkan tafsir tersebut bahwa Allah telah mengkaruniakan pendengaran, penglihatan dan hati agar manusia dapat berfikir akan tanda-tanda kebesaran Allah. Dalam hal ini Allah menciptakan sensor-sensor pada manusia yang mana dari pendengaran, penglihatan digunakan sebagai sensor untuk mendapatkan

informasi yang kemudian informasi itu akan dikelola dalam otak manusia yang disebut sebagai berfikir sehingga dalam berfikir terdapat suatu tahap pembelajaran yang kemudian otak akan memberikan respon terhadap informasi yang didapat dan sebagai penyaring dan penentu dari respon adalah hati nurani. Karena didalam hati nurani manusialah yang dapat membedakan mana yang baik dan buruknya respon yang dihasilkan.

Allah berfirman dalam surat Al Baqarah ayat 31 yang berbunyi:

وَعَلَّمَ آدَمَ الْأَسْمَاءَ كُلَّهَا ثُمَّ عَرَضَهُمْ عَلَى الْمَلَائِكَةِ فَقَالَ أَنْبِئُونِي بِأَسْمَاءِ هَؤُلَاءِ إِنْ كُنْتُمْ صَادِقِينَ

Artinya: Dan Dia mengajarkan kepada Adam nama-nama (benda-benda) seluruhnya, kemudian mengemukakannya kepada para Malaikat lalu berfirman: "Sebutkanlah kepada-Ku nama benda-benda itu jika kamu memang benar orang-orang yang benar".

Jalaluddin al-Mahalli & Jalaluddin as-Suyuthi Rohimahullah menjelaskan dalam kitab Tafsir Jalalain "(Dan diajarkan-Nya kepada Adam nama-nama) maksudnya nama-nama benda (kesemuanya) dengan jalan memasukkan ke dalam kalbunya pengetahuan tentang benda-benda itu (kemudian dikemukakan-Nya mereka) maksudnya benda-benda tadi yang ternyata bukan saja benda-benda mati, tetapi juga makhluk-makhluk berakal, (kepada para malaikat, lalu Allah berfirman) untuk memojokkan mereka, ("Beritahukanlah kepada-Ku) sebutkanlah (nama-nama mereka) yakni nama-nama benda itu (jika kamu memang benar.") bahwa tidak ada yang lebih tahu daripada kamu di antara makhluk-makhluk yang Kuciptakan atau bahwa kamulah yang lebih berhak untuk menjadi khalifah. Sebagai 'jawab syarat' ditunjukkan oleh kalimat sebelumnya."

Dalam hal ini secara tersirat Allah menunjukan dan mengajarkan nabi Adam tentang nama-nama benda sehingga dapat menyebutkan nama-nama benda yang telah Allah ajarkan kepada nabi Adam. Begitu pula dalam penelitian ini yang menggunakan sistem jaringan syaraf tiruan yang harus dilatih dahulu agar dapat mengenali objek kepala orang sehingga ketika ada sebuah gambar kerumunan yang diujikan maka, sistem sudah dapat mendeteksi dan menghitung banyaknya orang dalam gambar yang diujikan.

Dari penelitian ini dapat diambil pelajaran bahwa betapa kompleksnya Allah menciptakan sebuah sistem sebagaimana sistem syaraf manusia dalam mengelola informasi. Dengan mempelajari cara kerja sistem syaraf manusia dan mengaplikasikanya dalam membuat sebuah sistem syaraf tiruan untuk menghitung banyaknya orang dalam sebuah gambar. Sehingga dari sistem yang dibangun dapat dimanfaatkan salah satunya untuk menghitung banyaknya peserta yang tertarik dalam sebuah tema tertentu pada serangkaian workshop atau mungkin untuk kepentingan lain. Dengan begitu tidak perlu lagi repot-repot turun ke lapangan untuk menghitung langsung berapa banyak orang yang minat atau hadir dengan begitu akan lebih efektif dalam penyampaian informasi/ilmu.

يَلْغُوا عَنِّي وَلَوْ آيَةً

Artinya: “Sampaikanlah dariku walau hanya satu ayat” (HR. Bukhari)

Dalam hadis tersebut nabi menyeru umatnya untuk menyampaikan apa yang telah didapatkan walaupun sedikit. Karena orang yang menyampaikan ilmu tidak akan terputus pahalanya walaupun sudah meninggal sebagaimana sabda Rasulullah:

إِذَا مَاتَ الْإِنْسَانُ انْقَطَعَ عَمَلُهُ إِلَّا مِنْ ثَلَاثَةٍ مِنْ صَدَقَةٍ جَارِيَةٍ وَعِلْمٍ يُنْتَفَعُ بِهِ وَوَلَدٍ صَالِحٍ يَدْعُو لَهُ

Artinya: “*Jika seseorang meninggal dunia, maka terputuslah amalannya kecuali tiga perkara (yaitu): sedekah jariyah, ilmu yang dimanfaatkan, atau do’a anak yang sholeh*” (HR. Muslim)

Hadis tersebut menjelaskan salah satu keutamaan orang yang menyampaikan ilmu yaitu tidak akan terputus pahalanya walaupun sudah meninggal dunia karena sejatinya orang yang meninggal dunia sudah tidak bisa lagi berbuat apapun kecuali mempunyai 3 amalan yang disebutkan oleh nabi.

## **BAB V**

### **PENUTUP**

#### **5.1. Kesimpulan**

Telah dijabarkan pada bab-bab sebelumnya mengenai implementasi dan pengujian sistem *convolutional neural network* untuk menghitung banyaknya orang berdasarkan citra kepala maka, diperoleh kesimpulan bahwa algoritma tersebut cukup baik dalam memberikan hasil dalam mengidentifikasi dan menghitung banyaknya orang.

- 1) Pada tahap deteksi citra kepala pada gambar dilakukan pengujian akurasi menggunakan *5-fold cross validation* didapat hasil rata-rata dari *5-fold* sebesar 94.76% yang kemudian diujikan untuk mendeteksi 6 gambar yang terdiri dari 3 gambar dengan objek citra kepala orang dan 3 gambar *background* seperti perkebunan, kamar dan ruangan bermain anak-anak maka, didapati hasil 5 gambar terdeteksi dengan pernyataan yang sesuai (benar) dan 1 gambar dengan pernyataan yang salah.
- 2) pada tahap menghitung citra kepala pada gambar dengan menggunakan model sistem yang terbentuk dari deteksi citra kepala tadi pada 100 data gambar dengan objek perkumpulan orang sehingga didapat hasil yang cukup bagus yaitu sebesar 82% dengan 18 kesalahan perhitungan sistem dari 100 gambar yang diujikan.

#### **5.2. Saran**

Dalam pengembangan sistem menghitung banyaknya orang menggunakan algoritma *convolutional neural network* berdasarkan citra kepala orang maka perlu diperhatikan beberapa hal diantaranya:

1. Untuk mendapatkan hasil *bounding box* yang baik sebaiknya ditambahkan sebuah sistem lagi untuk mengatasi adanya *region* yang masih bertumpukan seperti *non-maximum suppression*.
2. Penelitian dapat dikembangkan lagi dengan memodifikasi arsitektur jaringan konvolusi sehingga dapat dibandingkan hasil dan pengaruh dari arsitektur yang dibangun.

## DAFTAR PUSTAKA

- Al-Mahalli, Jalaluddin & Jalaluddin As-Suyuthi. 2007. *Tafsir Jalalain*. Terj. Bahrun Abu Bakar. Jakarta: Sinar Baru Algensindo.
- Chen, D. Y., & Lin, K. Y. 2009. *A Novel Viewer Counter For Digital Billboards*. In 2009 Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing (pp. 653-656). IEEE.
- Chen, K. Loy, C. C. Gong, S. & Xiang, T. 2012. *Feature MINing For Localised Crowd Counting*. In *BMVC*. Vol. 1, No. 2, p. 3.
- Goodfellow, I., Bengio, Y., & Courville, A. 2016. *Deep Learning*. MIT press.
- <https://www.fritz.ai/object-detection/> (diakses pada tanggal 6 April 2020).
- <https://www.geeksforgeeks.org/selective-search-for-object-detection-r-cnn/> (diakses pada tanggal 14 April 2020).
- <https://www.geeksforgeeks.org/ml-stochastic-gradient-descent-sgd/> (diakses pada tanggal 16 April 2020).
- <https://ww2.mathworks.cn/en/discovery/object-detection.html> (diakses pada tanggal 6 April 2020).
- Ilyas, N., Shahzad, A., & Kim, K. 2020. *Convolutional-Neural Network-Based Image Crowd Counting: Review, Categorization, Analysis, and Performance Evaluation*. *Sensors*, 20(1), 43.
- Krizhevsky, A., Sutskever, I., & Hinton, G. E. 2012. *Imagenet Classification With Deep Convolutional Neural Networks*. In *Advances in neural information processing systems* pp. 1097-1105.
- LeCun, Y., Bengio, Y., & Hinton, G. 2015. *Deep Learning*. *Nature*, 521(7553), 436-444.
- M. Quraish Shihab, *Tafsir al-Misbah*, Jakarta : Lentera Hati, 2012.
- Park, Y. S., & Lek, S. 2016. *Artificial Neural Networks: Multilayer Perceptron for Ecological Modeling*. In *Developments in environmental modelling*. Vol. 28, pp. 123-140. Elsevier.
- Pham, V. Q., Kozakaya, T., Yamaguchi, O., & Okada, R. 2015. *Count Forest: Co-Voting Uncertain Number Of Targets Using Random Forest For Crowd Density Estimation*. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 3253-3261).

- Sena, S. 2018. *Pengenalan Deep Learning Part 8: Gender Classification Using Pre-Trained Network*. <https://medium.com/@samuelsena/pengenalan-deep-learning-part-8-gender-classification-using-pre-trained-network-transfer-37ac910500d1> (diakses pada tanggal 3 April 2020).
- Sindagi, V. A. & Patel, V. M. 2017, *Generating High-Quality Crowd Density Maps Using Contextual Pyramid CNNs*, in 'IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017', IEEE Computer Society, pp. 1879–1888.
- Uijlings, J. R., Van De Sande, K. E., Gevers, T., & Smeulders, A. W. 2013. *Selective Search For Object Recognition*. *International Journal Of Computer Vision*, 104(2), 154-171.
- Wang, C., Zhang, H., Yang, L., Liu, S., & Cao, X. 2015. *Deep People Counting In Extremely Dense Crowds*. In *Proceedings Of The 23rd ACM International Conference on Multimedia* (pp. 1299-1302).
- Wang, J., Wang, L., & Yang, F. (2017, March). *Counting Crowd with Fully Convolutional Networks*. In *2017 2nd International Conference on Multimedia and Image Processing (ICMIP)* (pp. 210-214). IEEE.
- Zhang, C., Li, H., Wang, X., & Yang, X. 2015. *Cross-Scene Crowd Counting Via Deep Convolutional Neural Networks*. In *Proceedings Of The Ieee Conference on Computer Vision And Pattern Recognition* (pp. 833-841).